C/C++ Development Kit

for

TERN 16/32-bit embedded controllers

with

Paradigm C++-TERN Edition

Technical Manual

1724 Picasso Ave. Suite A, Davis, CA 95616, USA
Tel: 530-758-0180 Fax: 530-758-0181
Internet Email: sales@tern.com ***http://www.tern.com***

Version 1.00

February 2, 2000

© 2000 TERN INC.
1724 Picasso Ave., Suite A, Davis, CA 95616, USA
Tel: 530 758 0180 Fax: 530 758 0181
Internet Email: sales@tern.com ***http://www.tern.com***

Important Notice

TERN is developing complex high technology integration systems. These systems are integrated with software and hardware which are not 100% defect free. ***TERN products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications. TERN*** and the Buyer agree that ***TERN*** will not be liable for incidental or consequential damages arising from the use of ***TERN*** products. It is the Buyer's responsibility to protect life and property against incidental failure.

TERN reserves the right to make changes and improvements to its products without providing notice.

TERN will be responsible for technical support of this kit, NOT Borland. Do not contact Borland International for technical support assistance.

Table of Contents

Chapter 1: Introduction

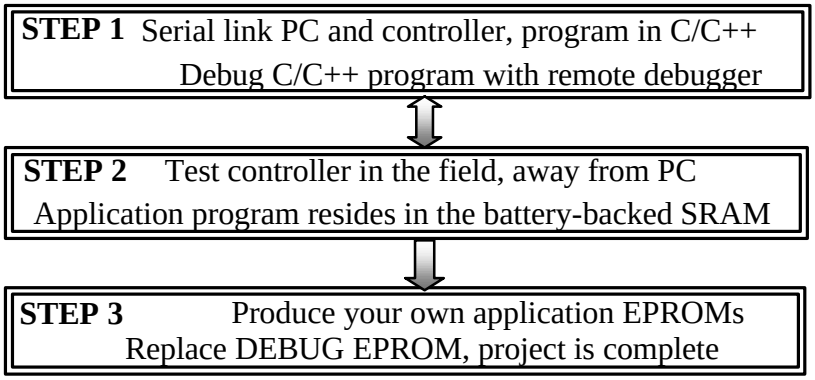
1.1 Description

Welcome to TERN C/C++ Evaluation (EV) or Development Kit (DV) for 16/32-bit embedded controllers. TERN controllers are based on PC compatible processors, easy to program in C/C++. The use of ANSI C/C++ compilers for your projects will support your development needs now and into the future.
TERN's EV/DV Kit supports a complete user-friendly integrated development environment for embedded application. It integrates Paradigm C++-TERN Edition, a state-of-the-art integrated development environment (IDE), including a C/C++ compiler, remote debugger, C libraries, I/O drivers, configuration files, project files, sample programs, technical manuals, power supply, serial cable, remote debug kernel, and a TERN controller.

Within the IDE, you can create, debug, and deploy real-time embedded system applications without resorting to the use of external tools. If you are used to running separate editor, debugger, make, and other tools to get a job done, then you are in for a real treat with Paradigm C++-TERN Edition. After installing the software and serial link connecting your PC with the controller, you can run a sample program "led.c" by simply click the "run" button. The kit takes care of the whole operation steps including compile, link, locate, download, and ready to run. You can single step, set breakpoint, or run your program on the target controller. After debug and setting a jumper on the board, you can take the controller to a standalone field test. After the standalone field test, the DV kit can generate a HEX/BIN file for your EPROM/Flash. It will dramatically reduce your development time and tool investments. You'll be able to put a dedicated system together in record time.

TERN EV/DV kits support a complete product line of 16/32-bit embedded controllers based on AM188/186, i386EX, and V25 processors, featuring operational amplifiers, ADC, DAC, real time clock, EE, solenoid drivers, power relays, UART, opto-coupler, LCD, keypads, PCMCIA, Flash, 9-bit network, and quadrature decoders. MemCard-A™ can support up to 1 GB external memory space. With ACTF™ support, you can download a program to a FLASH without burning a ROM.

1.2 Development of application software consists of just three easy steps



1.2.1 STEP 1

You can program TERN controllers from your PC via serial link with an RS232 interface. Your C/C++ program can be remotely debugged over the serial link at up to 115,000-baud rate. You can single-step, set

breakpoint, or run your program. Figure 1.1 illustrates how the A-Engine, VE232, serial cable and PC are connected.

1.2.2 STEP 2

After debugging your program, changing a single jumper will permit you to test-run the controller in the field, away from the PC, with the application program residing in battery-backed SRAM. The battery should last 3-5 years under normal conditions.

1.2.3 STEP 3

When the STEP 2 field test is complete, an application HEX or BIN file can be easily generated on your PC. You may produce your own ROM, or download the application file into FLASH.

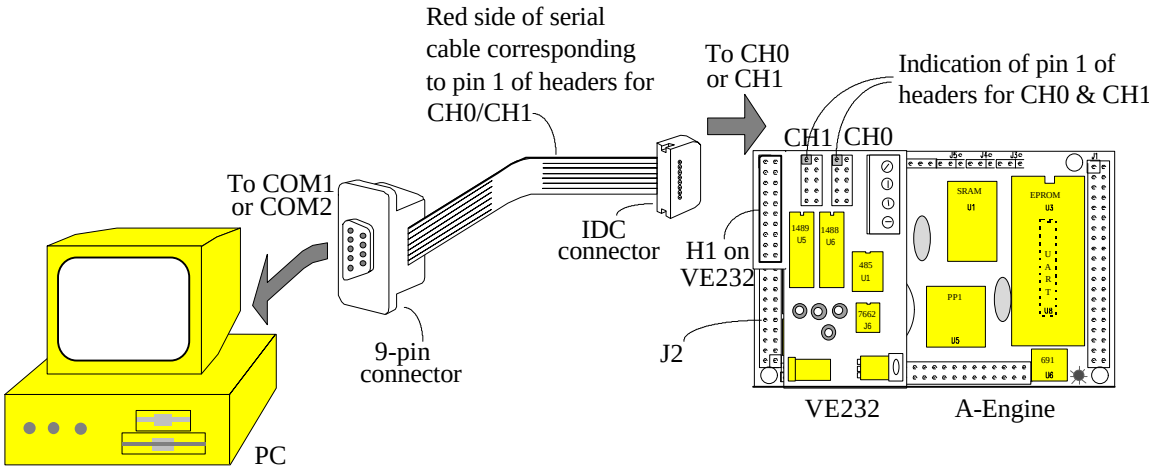


Figure 1.1 Hardware Installation of A-Engine™, VE232™, Serial Cable, and a PC

Chapter 2: Installation

2.1 Software Installation

Installation instructions launch automatically from the Paradigm C++-TERN Edition CD.

- If your are using V25 based controller, your working directory is C:\TERN\V25
- If your are using Am186/188 based controller, your working directory is C:\TERN\186
- If your are using i386EX based controller, your working directory is C:\TERN\386

2.2 Hardware Installation

Use A-Engine as an example. Install the VE232™ interface with the H1(10x2) socket connector on the upper half of the J2 (dual row header) of the A-Engine™. Figure 2.1a and 2.1b shows the VE232™ and the A-Engine™ before and after installation. A serial cable (PC-V25) should be used to link the A-Engine™ with your PC. Install the 5x2 IDC connector to the V232™ CH0 header (J2). *Note that the red side of the cable must point to pin 1 of the VE232 J2 header. A small circle is drawn for indication in the Fig. 2.2.* The DB9 connector should be connected to the PC's COM1/COM2. Connect a wall transformer +9V DC output to the VE232™ DC power jack. The LED should blink twice after the A-Engine™ is powered off/on or reset (Figure 2.3).

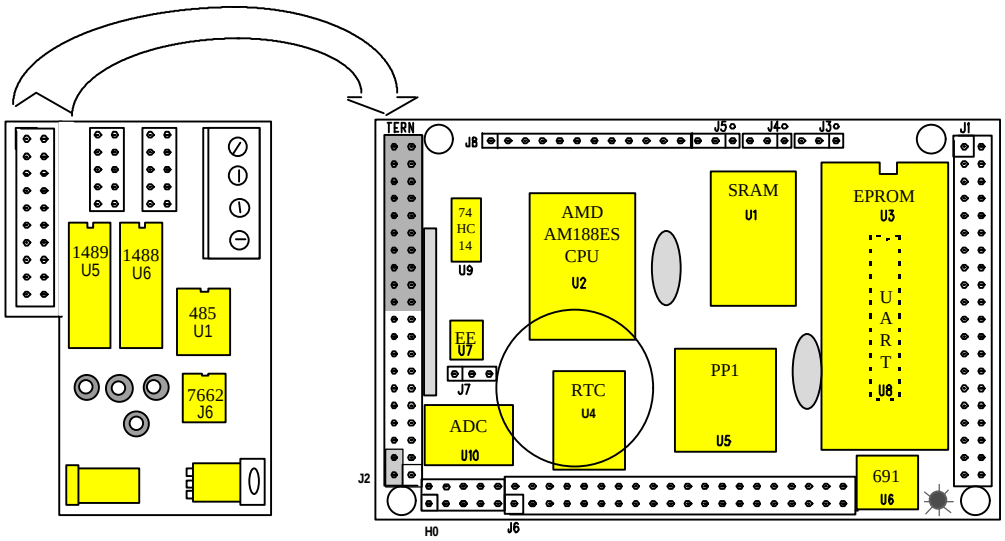


Figure 2.1a. Before installing the VE232™ on the A-Engine™

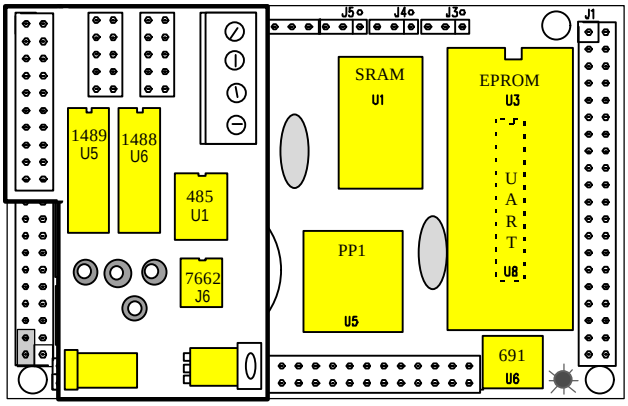


Fig. 2.1b. After installing the VE232™ on the A-Engine™

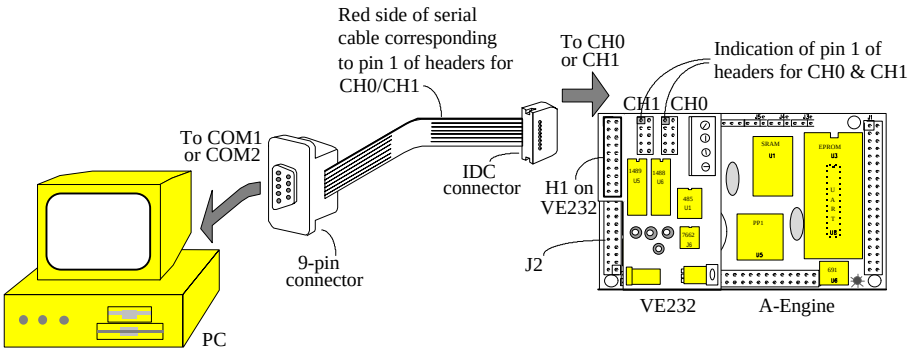


Fig. 2.2. Serial connection between the A-Engine™ and the PC

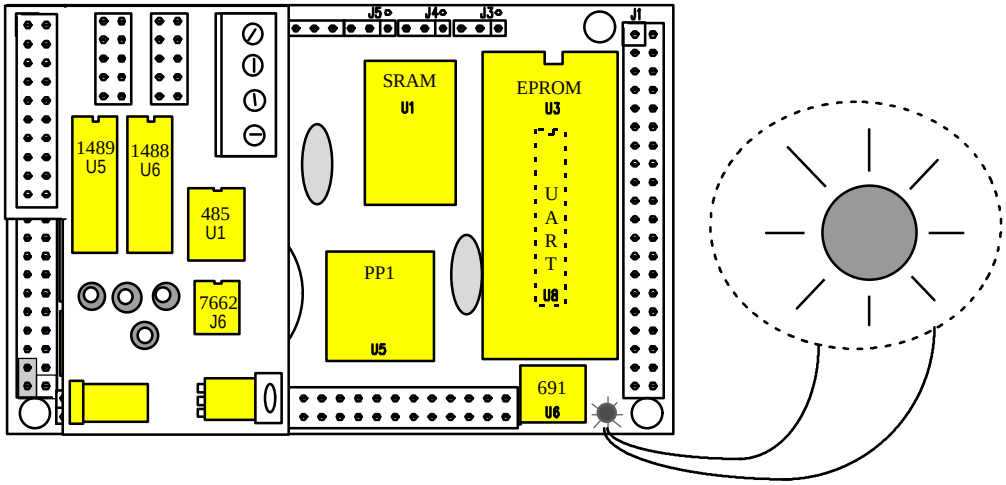
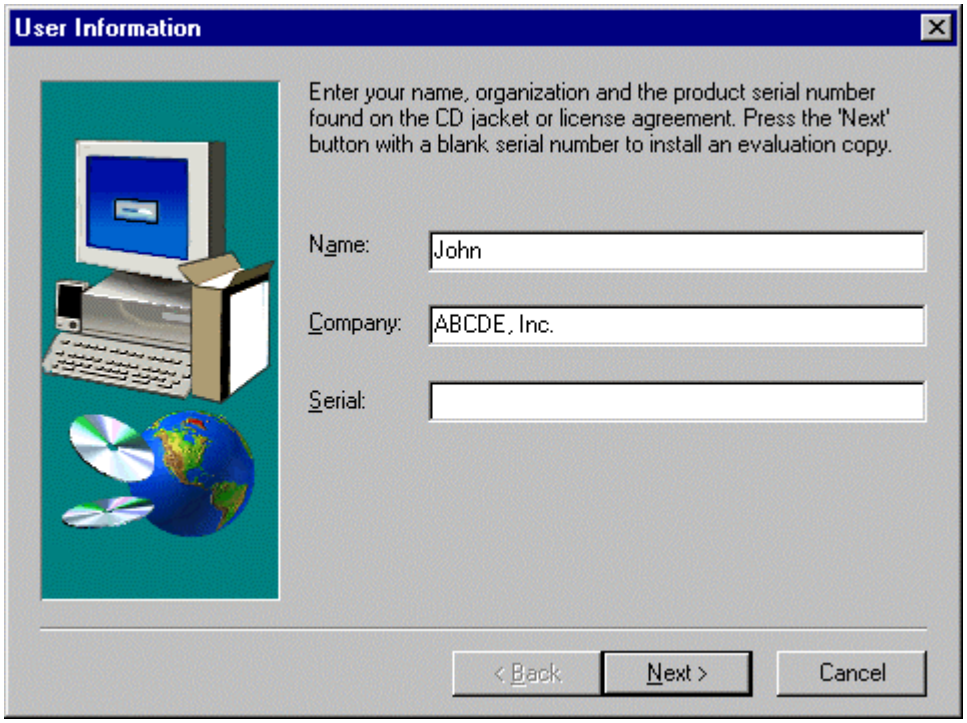


Fig. 2.3. The LED blinks twice indication hardware ready.

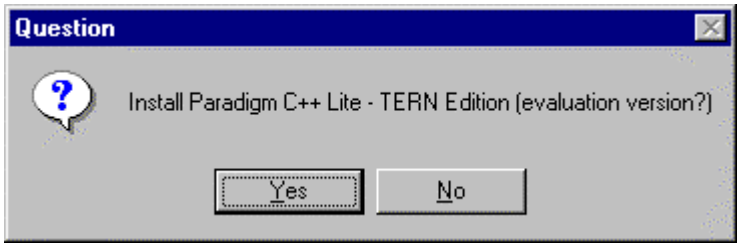
Quick Start Guide

Getting started

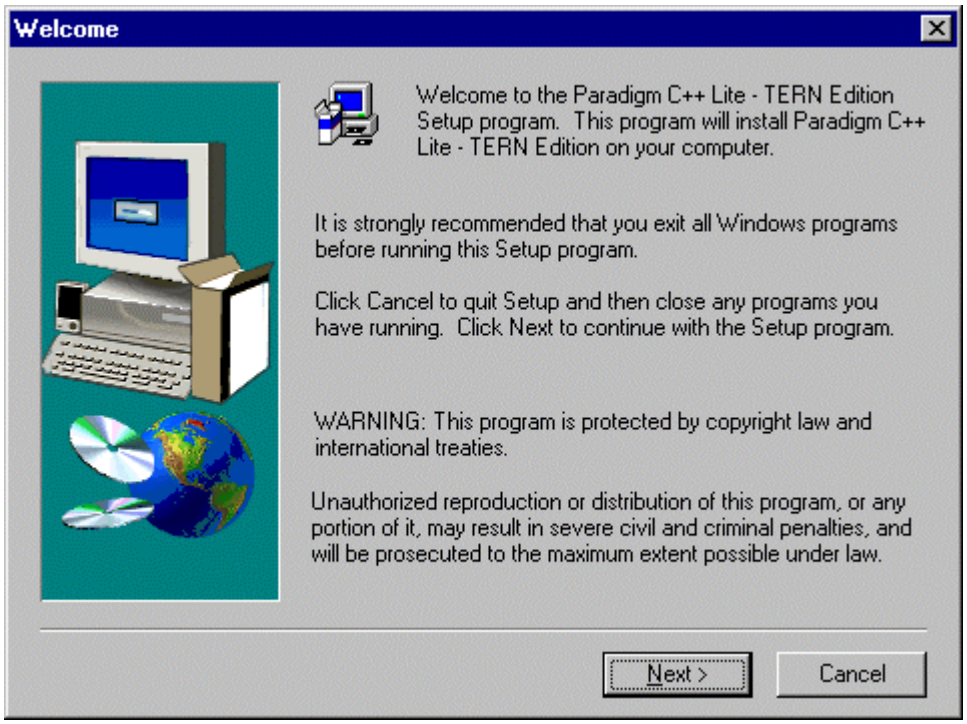
Installation instructions launch automatically from the TERN CD. It will prompt you as shown below.



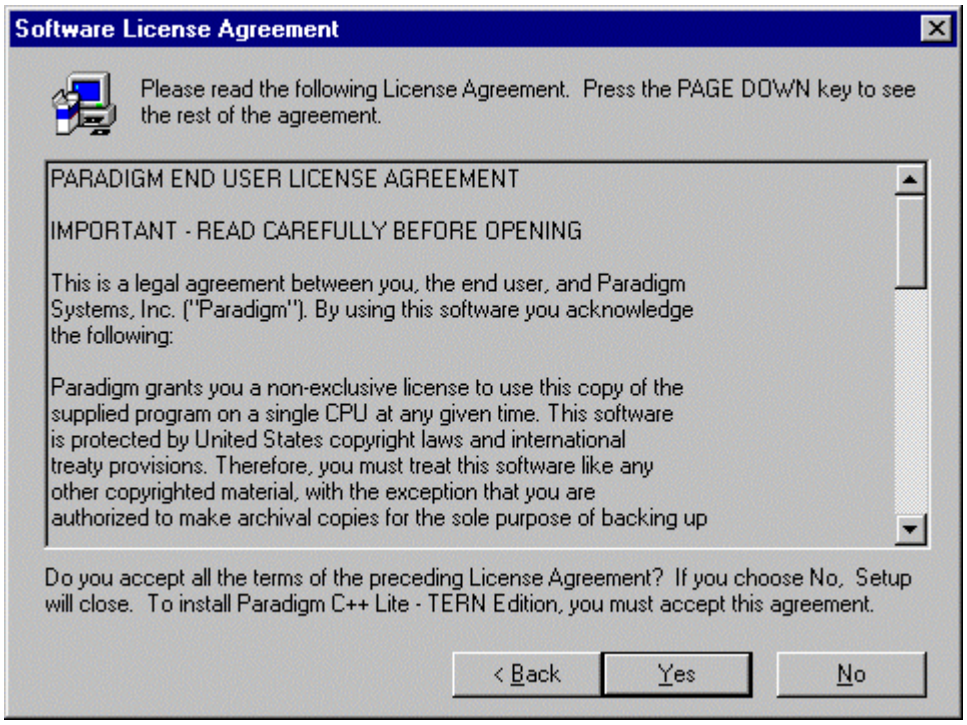
If you purchased a full version DV-P kit from TERN, you will have a Serial Number from TERN with your order. Type in your Serial number for the full version installation. If you do not have a correct serial number, you can install the Paradigm C++ Lite-TERN Edition evaluation version, by pressing the “Next” button without any serial number.



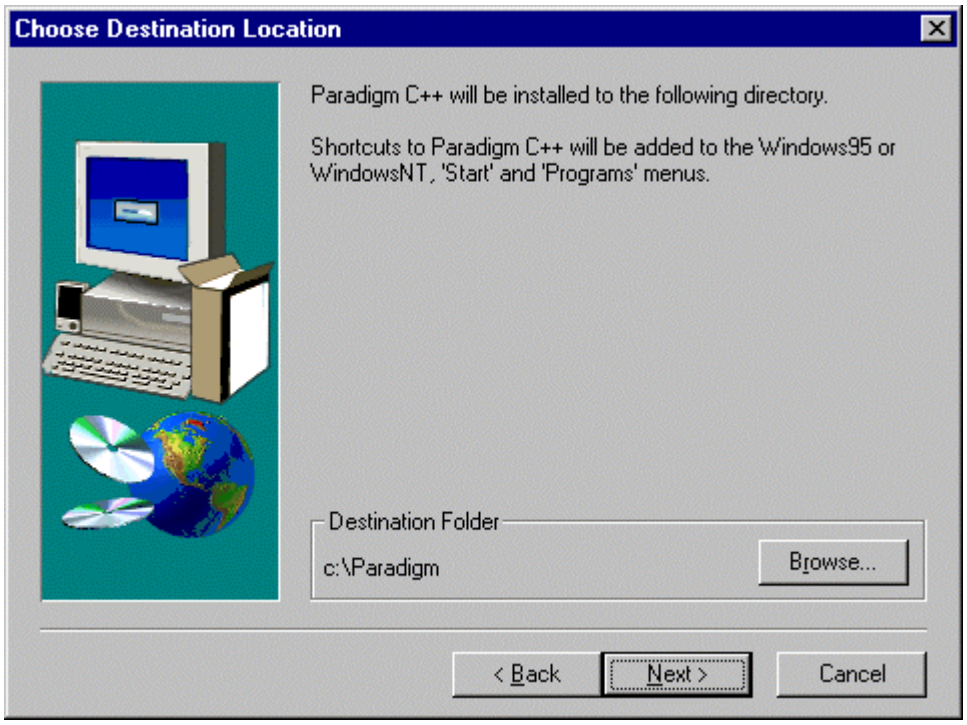
“Yes”, then “Next”.


Welcome
Welcome to the Paradigm C++ Lite - TERN Edition Setup program. This program will install Paradigm C++ Lite - TERN Edition on your computer.
It is strongly recommended that you exit all Windows programs before running this Setup program.
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.
WARNING: This program is protected by copyright law and international treaties.
Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.
Next >
Cancel

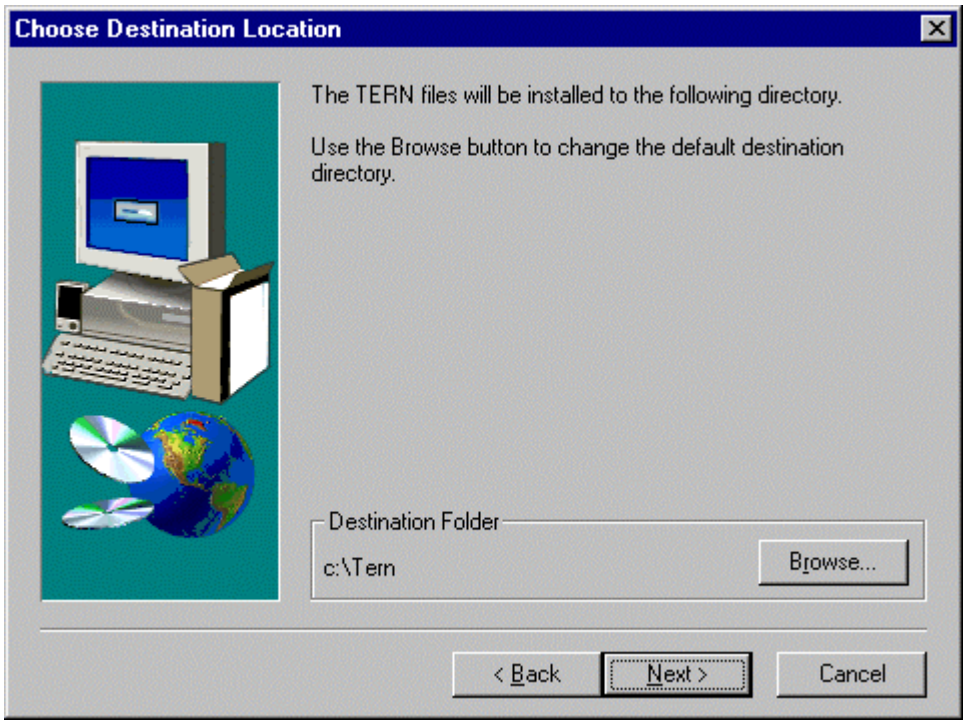
"Next"


Software License Agreement
Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.
PARADIGM END USER LICENSE AGREEMENT
IMPORTANT - READ CAREFULLY BEFORE OPENING
This is a legal agreement between you, the end user, and Paradigm Systems, Inc. ("Paradigm"). By using this software you acknowledge the following:
Paradigm grants you a non-exclusive license to use this copy of the supplied program on a single CPU at any given time. This software is protected by United States copyright laws and international treaty provisions. Therefore, you must treat this software like any other copyrighted material, with the exception that you are authorized to make archival copies for the sole purpose of backing up
Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Paradigm C++ Lite - TERN Edition, you must accept this agreement.
< Back
Yes
No

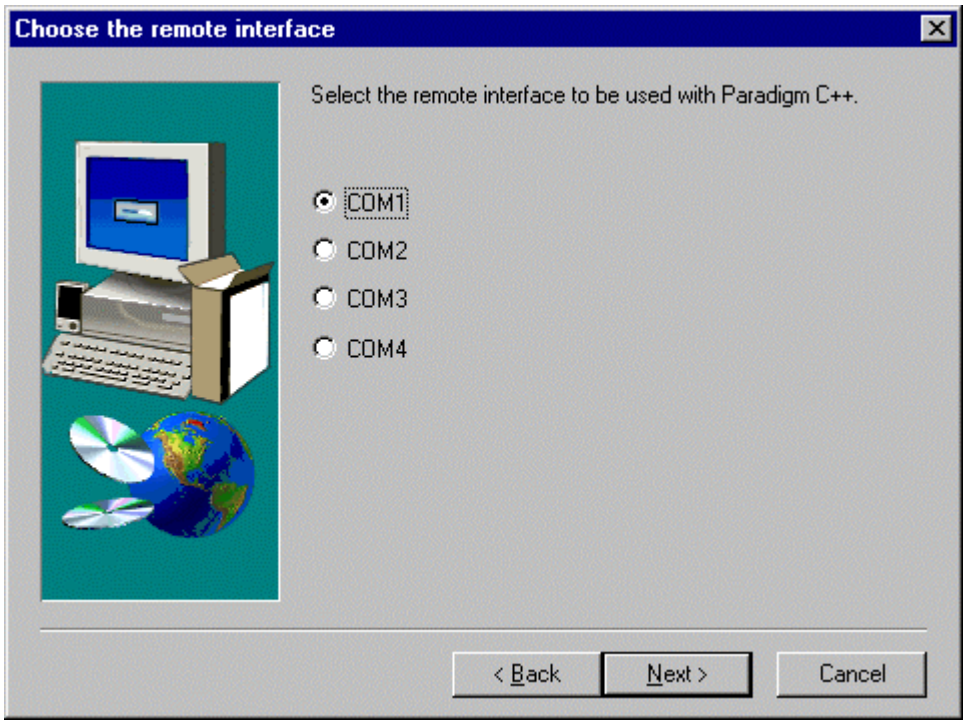
"Yes"


Choose Destination Location
Paradigm C++ will be installed to the following directory.
Shortcuts to Paradigm C++ will be added to the Windows95 or WindowsNT, 'Start' and 'Programs' menus.
Destination Folder
c:\Paradigm
Browse...
< Back
Next >
Cancel

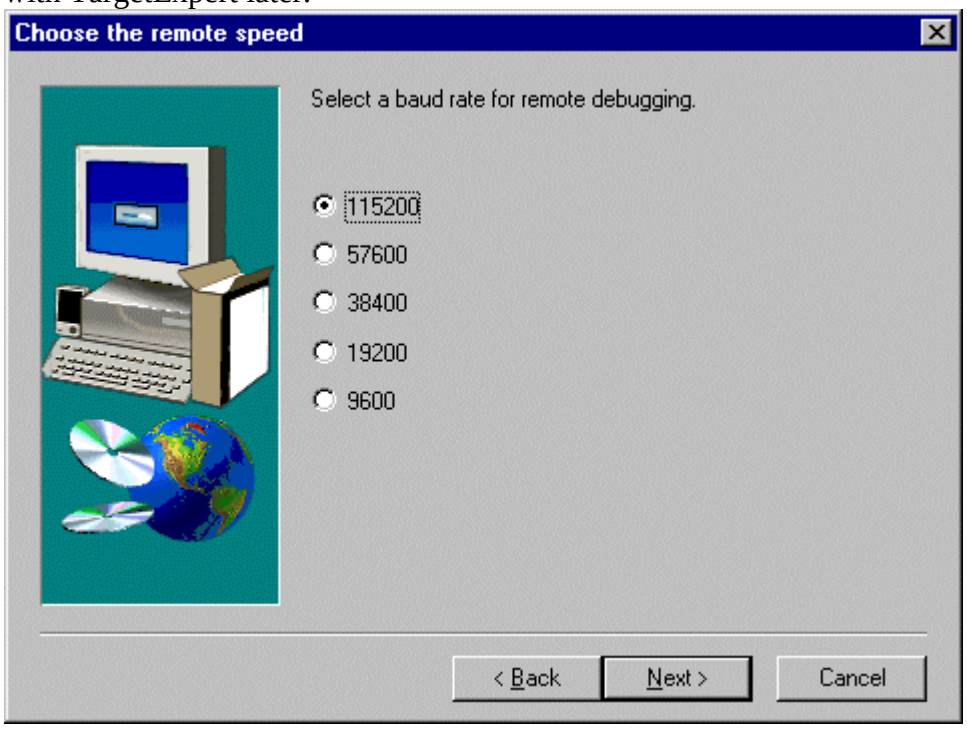
"Next"


Choose Destination Location
The TERN files will be installed to the following directory.
Use the Browse button to change the default destination directory.
Destination Folder
c:\Tern
Browse...
< Back
Next >
Cancel

"Next"



Select “COM1” for example, then “Next”.
If you are not sure which COM port you will use, you can modify the COM port setting with TargetExpert later.

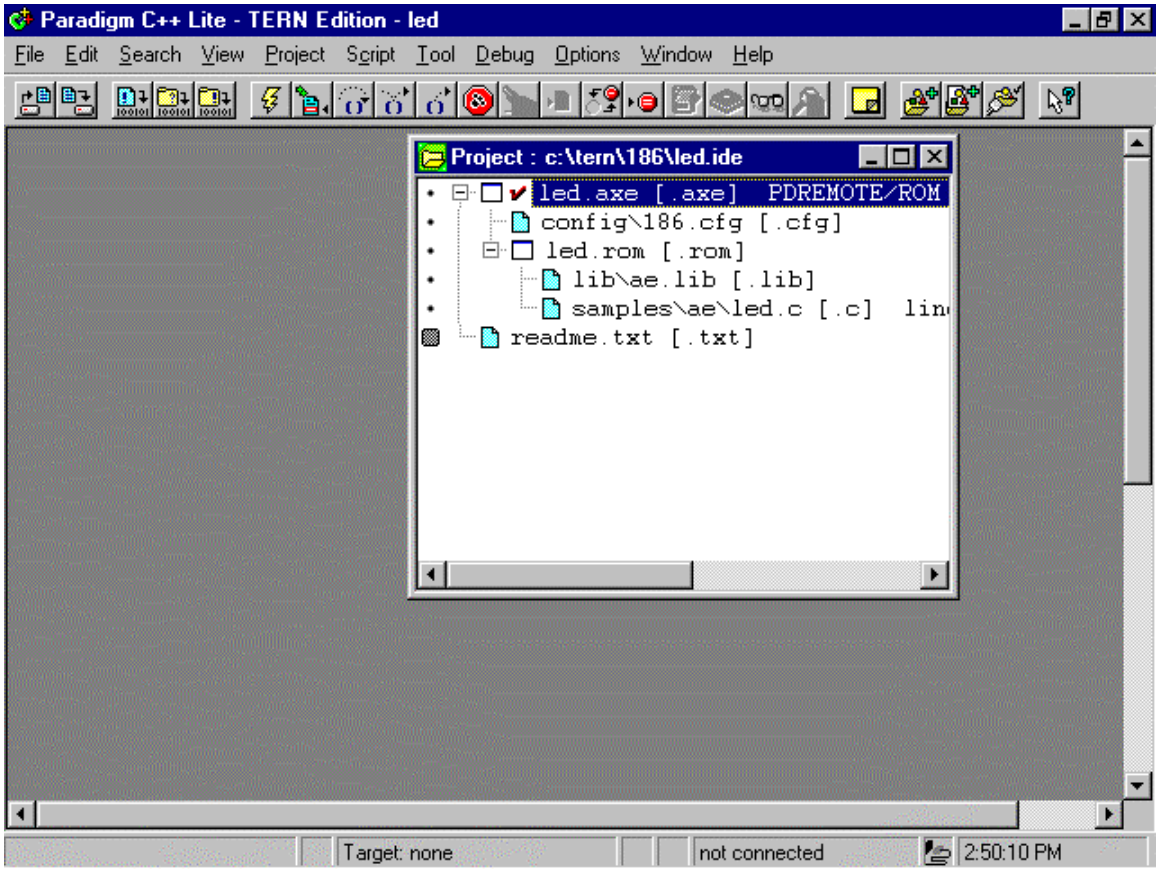


Select 115,200 for example.
Please also check your TERN controller with the corresponding DEBUG ROM installed. By default, most TERN controllers are setup for 115,200 baud remote debugging. You can change this baud rate later.

After following the instructions, “Finish”, exit the install screen, a program item titled Paradigm C++-TERN Edition will be added to the Windows95/98/NT/2000 “Start” menu.

Double click the icon of Paradigm C++-TERN Edition to launch the IDE.

At the first time, a default c:\tern\186\LED.IDE project will be open for TERN’s AM188/186 based controllers.



If you have a TERN controller connecting to PC COM1/2 via RS232 cable, it will show a downloading window waiting for the remote controller to response. If the remote controller works with a DEBUG ROM installed (for example, lable AE_0_115, see DEBUG ROM list included in the CD terndocs directory), application code will be downloaded, otherwise will show a time out prompt.

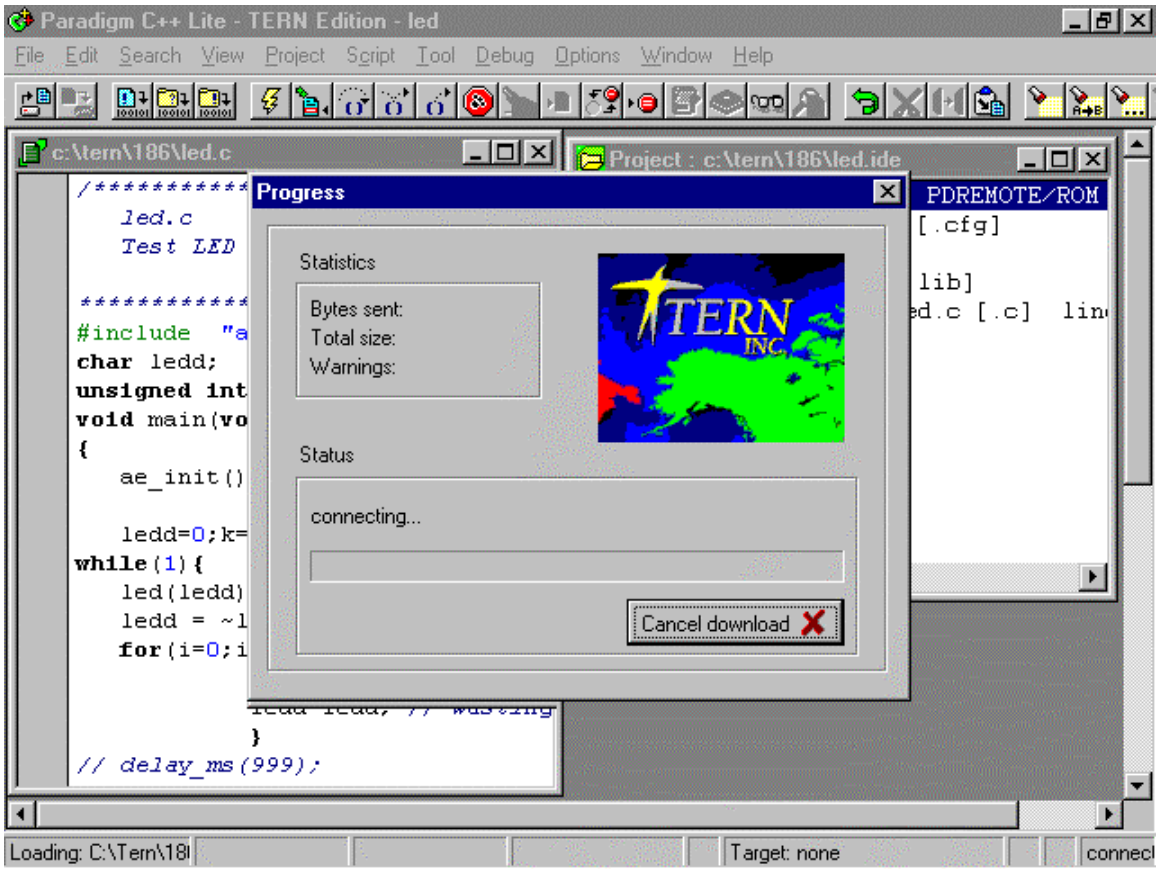
If you have a connecting problem, You should check:

1) The RS-232 connection, the Cable red strip should pointing to the pin 1 of the 10-pin header. When the DC power on your controller, a red LED should blink twice and then stay on.

2) The DEBUG ROM has a correct label. See CD for DEBUG_ROM_list. The default debug baud rate is 115,200, if you are using a 40 MHz A-Engine, RS-232 cable connecting to SER0, with a DEBUG ROM "AE_0_115".

3) For a 20 MHz A-Engine, you must change the debug baud rate to 57,600. If you want to change the Baud Rate setting, use "TargetExpert".

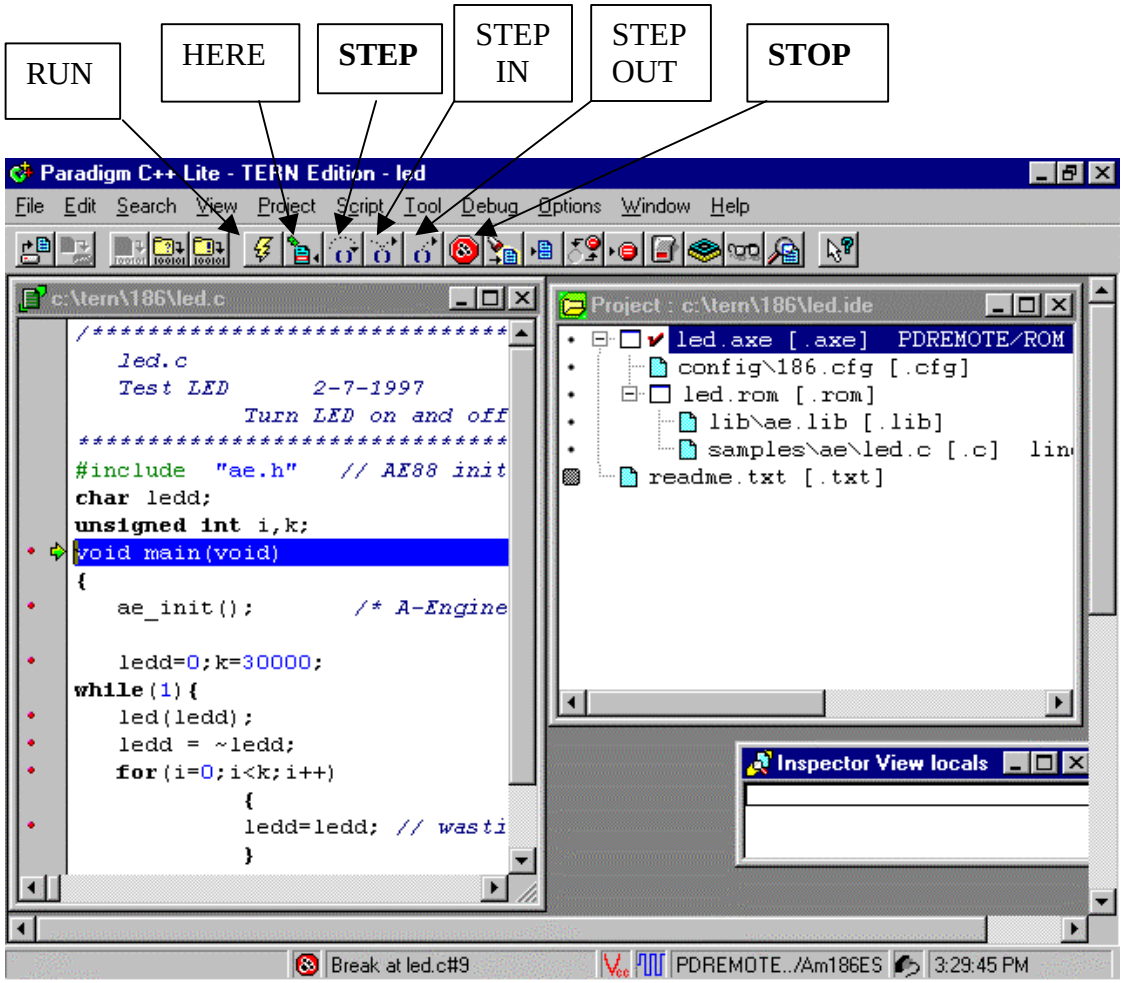
4) Your PC connection is using COM1 by a default installation, otherwise you have to change the COM port setting with "TargetExpert".



With no connection, the bottom of the window shows "Target none".

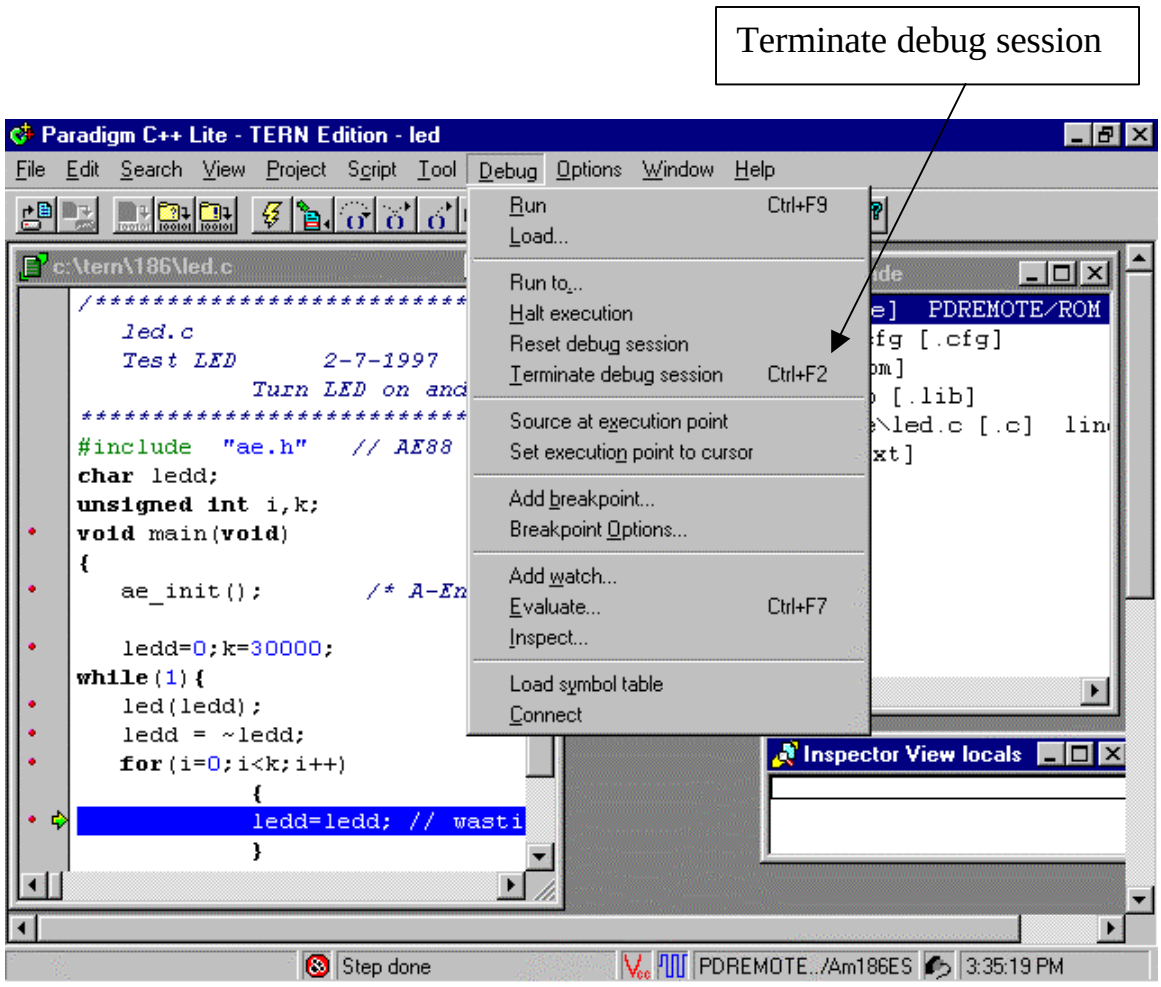
After your code is successfully downloaded, a debug window with C source code will show up with a blue highlight line at the C code main() line, indicating the code stop at that line. The bottom of the window will show "Break at led c#9", a Vcc and red waveform indicating ready for you to debug and run.

You can use mouse left click at the "Single Step" button at the top SpeedBar, or "F8" on keyboard to single step your code. Use mouse to select "Run", then the LED on your controller should blink. Hit the RED button to "Stop".

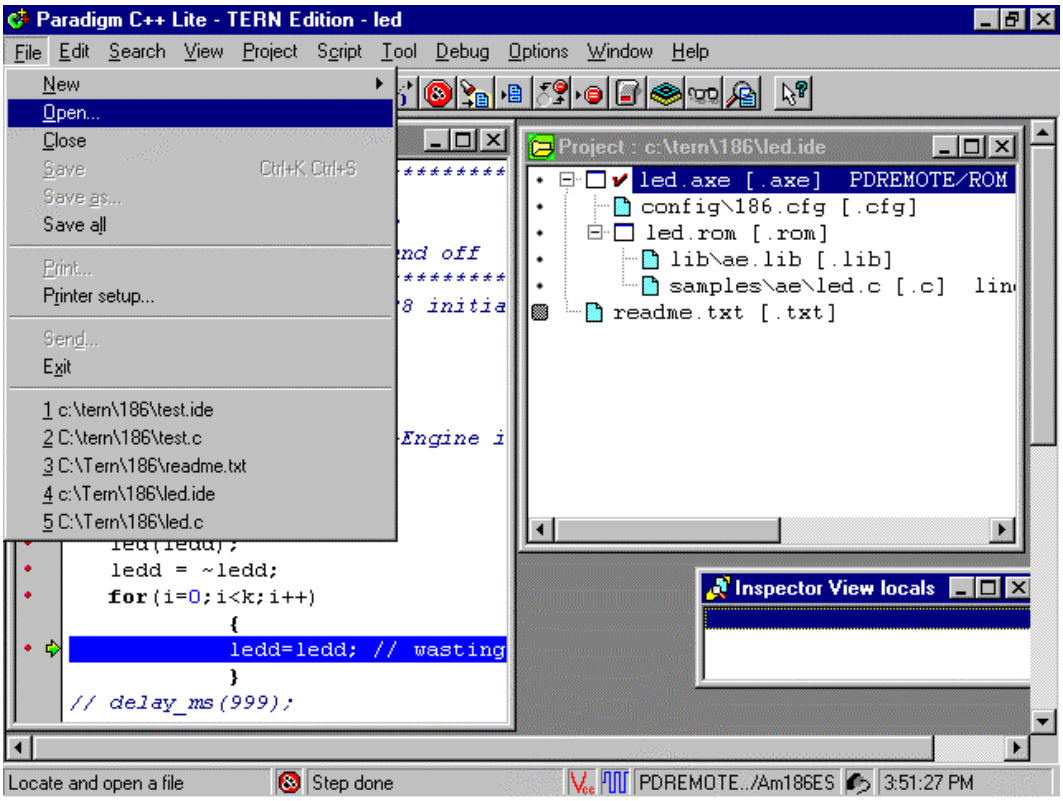

RUN
HERE
STEP
STEP IN
STEP OUT
STOP
Paradigm C++ Lite - TERN Edition - led
File Edit Search View Project Script Tool Debug Options Window Help
c:\tern\186\led.c
/*****************************
led.c
Test LED 2-7-1997
Turn LED on and off
****
#include "ae.h" // AE88 init
char ledd;
unsigned int i,k;
void main(void)
{
ae_init(); /* A-Engine
ledd=0;k=30000;
while(1){
led(ledd);
ledd = ~ledd;
for(i=0;i<k;i++)
{
ledd=ledd; // wasti
}
Project : c:\tern\186\led.ide
led.axe [.axe] PDREMOTE/ROM
config\186.cfg [.cfg]
led.rom [.rom]
lib\ae.lib [.lib]
samples\ae\led.c [.c]
readme.txt [.txt]
Inspector View locals
Break at led.c#9
PDREMOTE../Am186ES
3:29:45 PM

How do I terminate the debug session ?

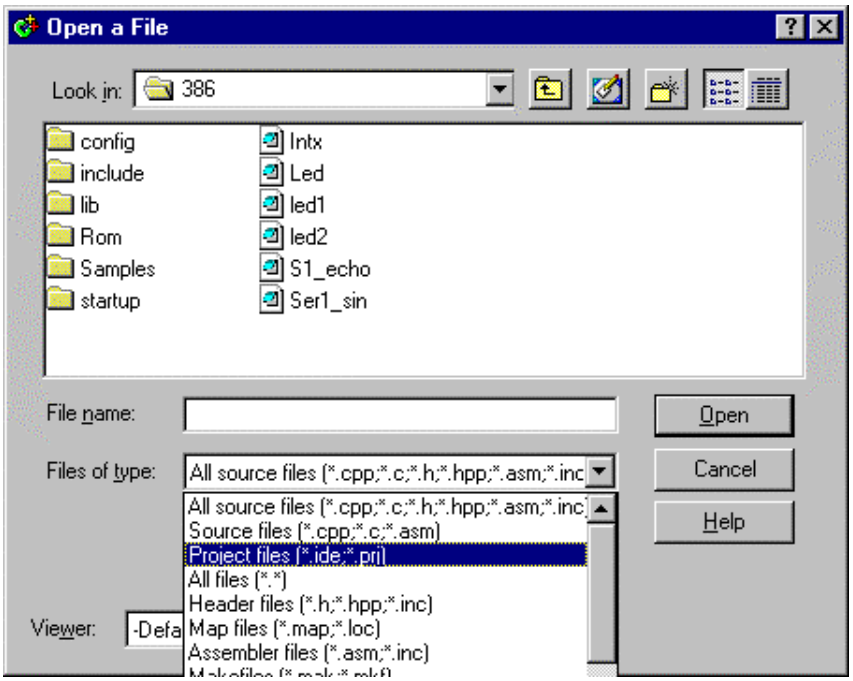
Select the top SpeedBar “Debug” menu, right click on “Terminate debug session”, or “Ctrl+F2” key. The led on your controller should blink twice then stay on indicating a RESET. Then you can stop your debug session, close the IDE.

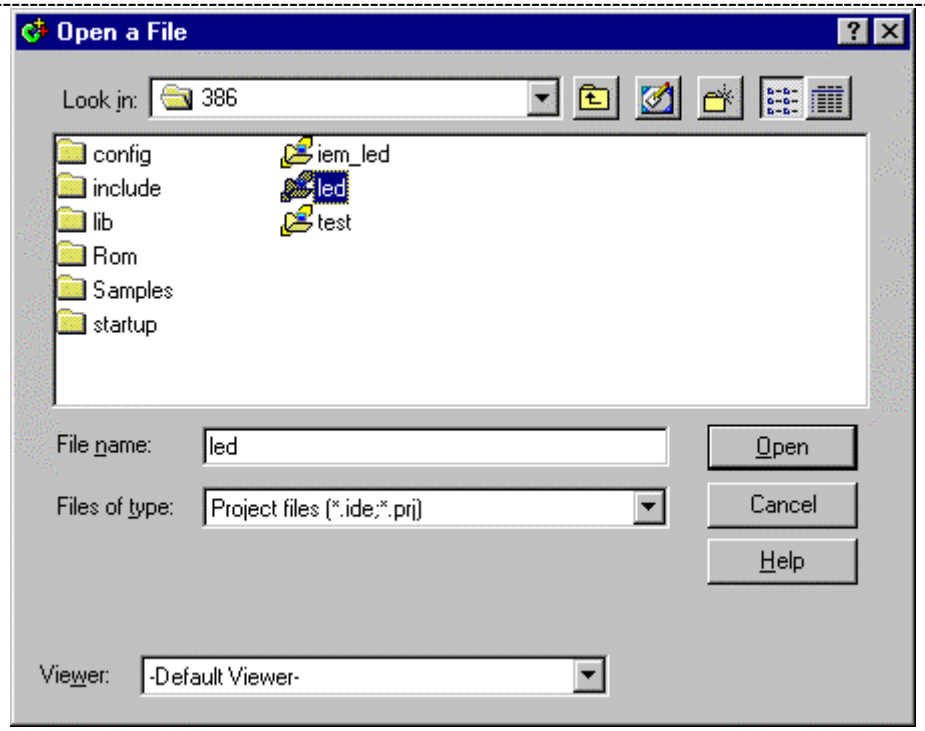


How do I open an existing project ?

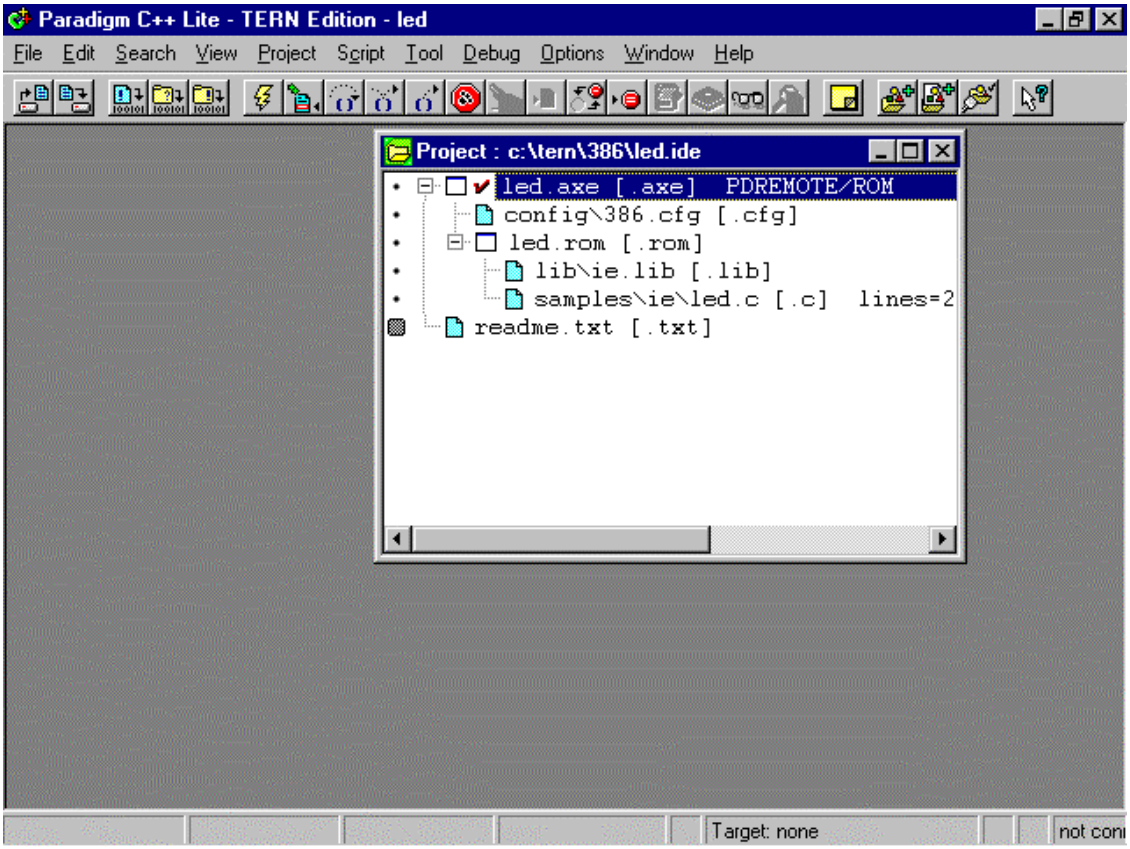


If you have an i386EX, or a V25 based controller, you can not use the c:\tern\186\led.ide project, you need to open another project c:\tern\386\led.ide, or c:\tern\V25\led.ide.
Use i386-Engine as example. The working directory is c:\tern\386.
Try to open a new project file. You need to select the existing led.ide file, under c:\tern\386 directory.



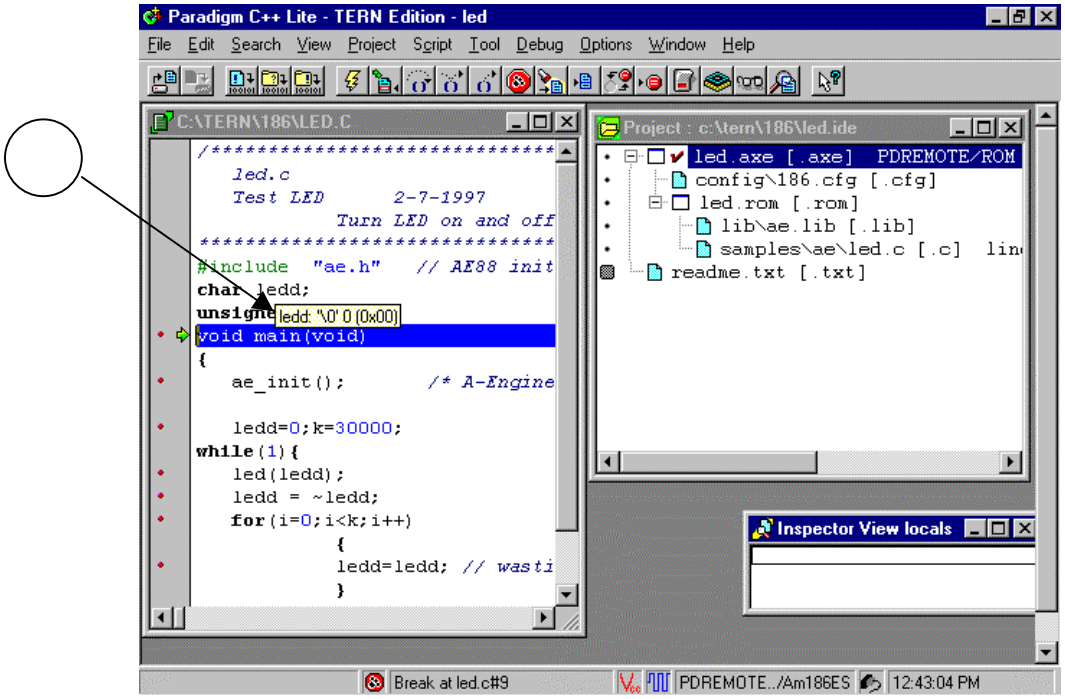


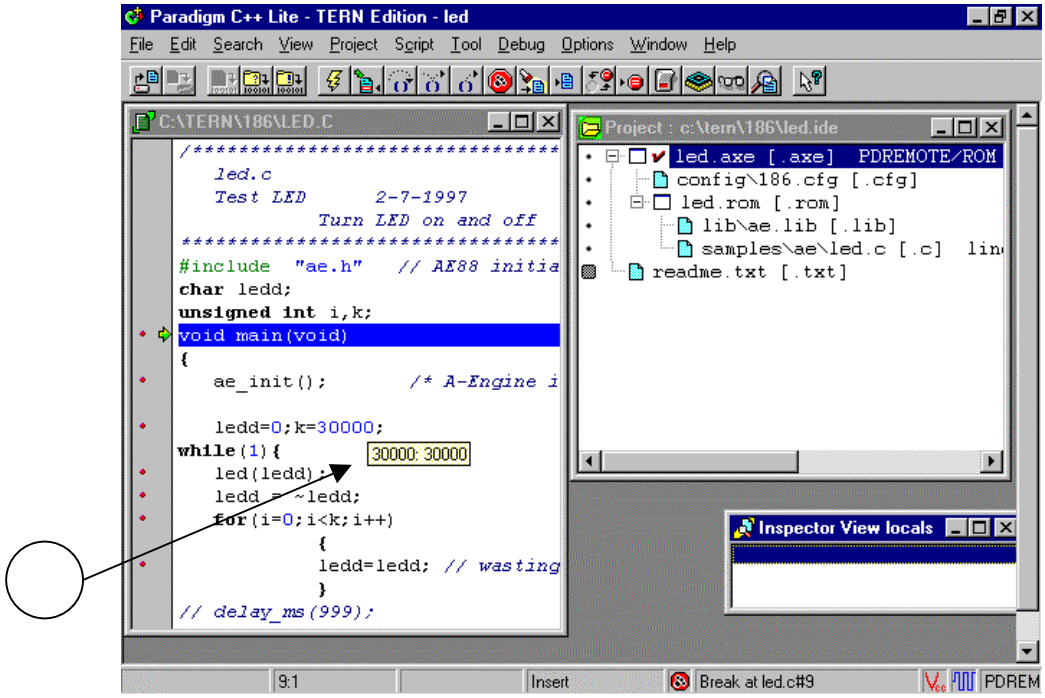
Select “Open”, then a project window for the 386 will be available.



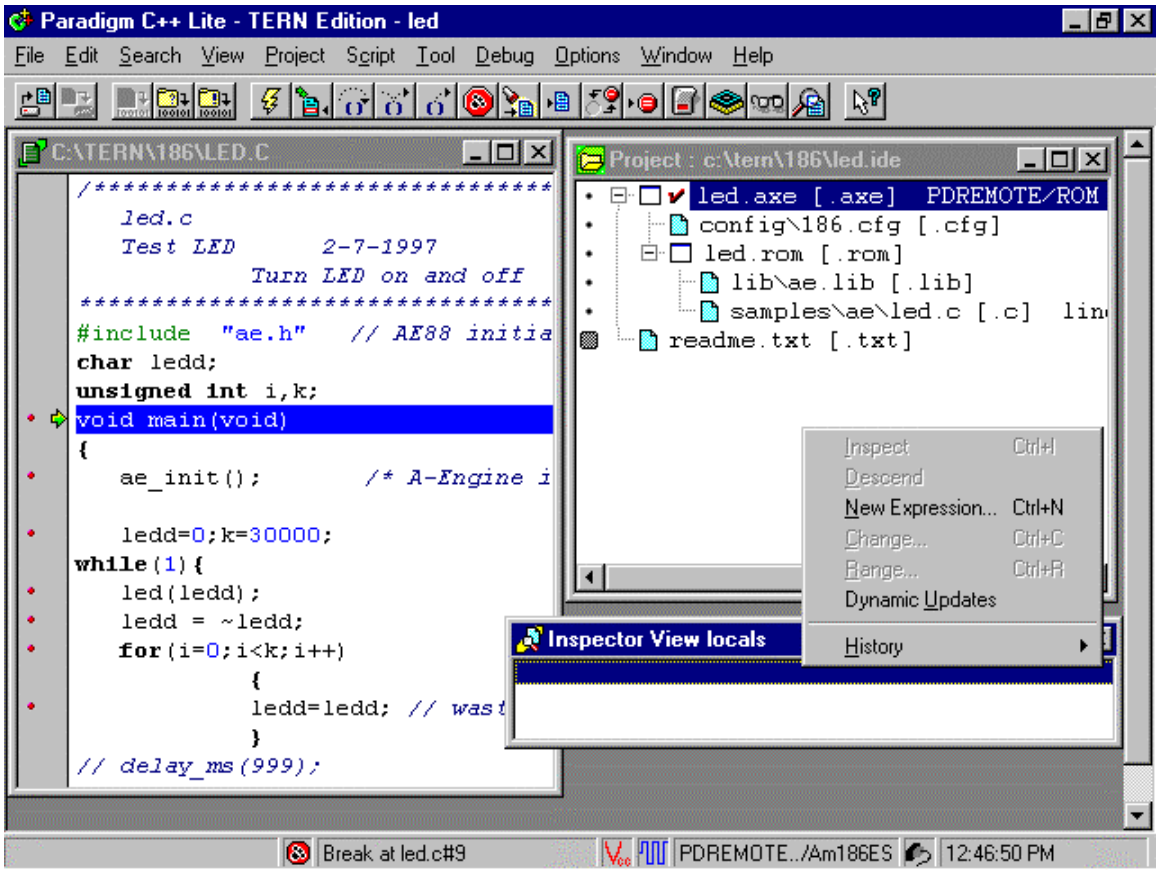
How do I inspect the value of a variable ?

After downloading the code or during debugging, if you want to see what value of the variable “ledd” or “k”, you can simply use mouse point to the “ledd” or “k”, a report will be shown.

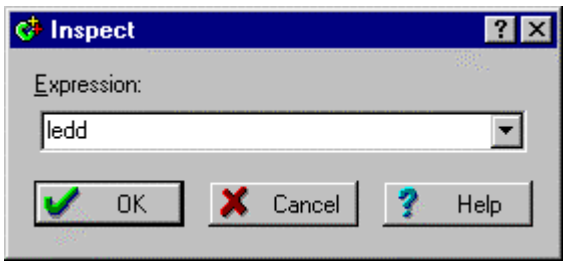




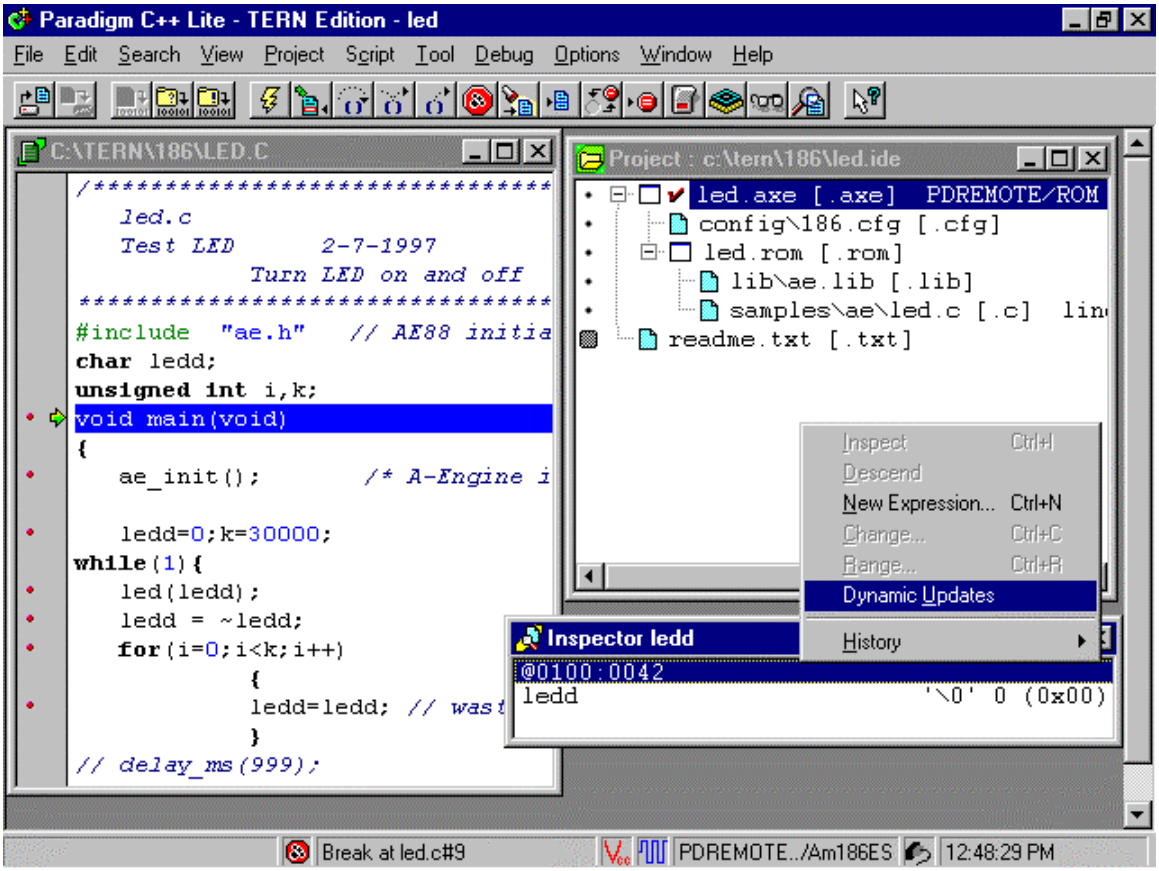
You can also right click in the “Inspector View” window. A menu popup show “New Expression” or “Dynamic Updates”.



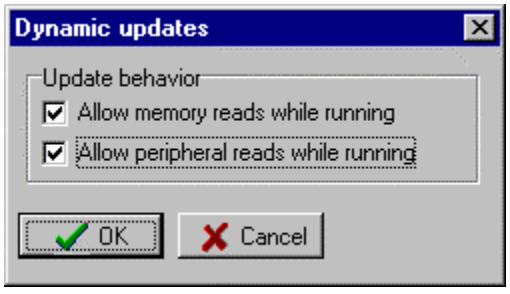
If you are interested on “ledd”, you enter the “ledd” to add it into “Inspector View” window.



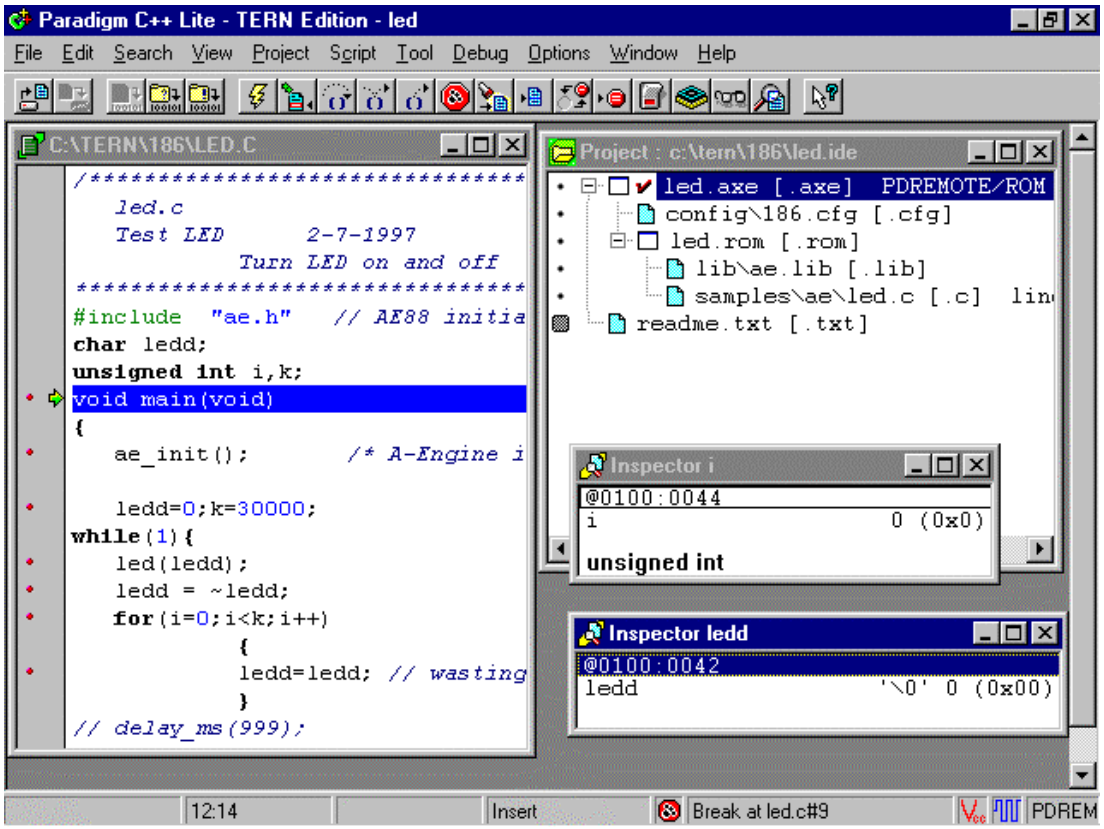
Then “OK”.



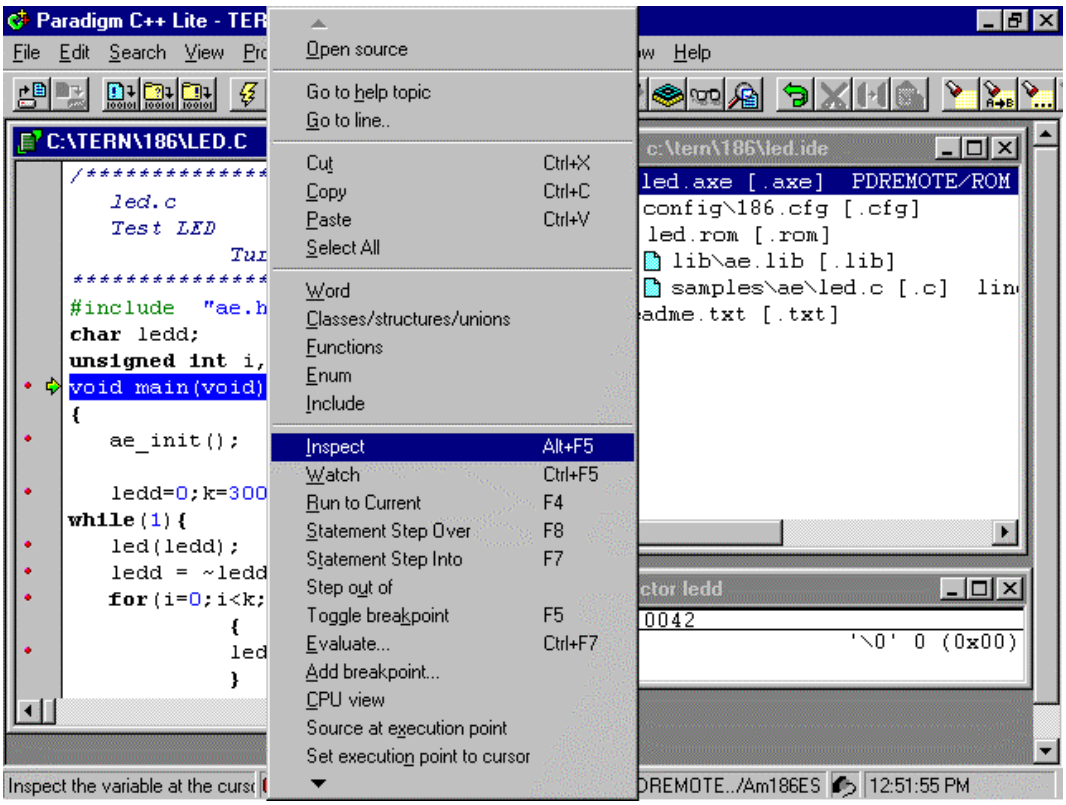
The ledd will be seen in the “Inspector” window. You may enable the “Dynamic Updates” by right click on the “Inspector” window, select the “Dynamic Updates”. Please be aware of the interrupt latency and communication overhead while the “Dynamic updates” enabled.



After the Dynamic updates being enabled, you can see the value change in the Inspector windows.
Disable the Dynamic updates, if you want to test your own code for a real time performance.

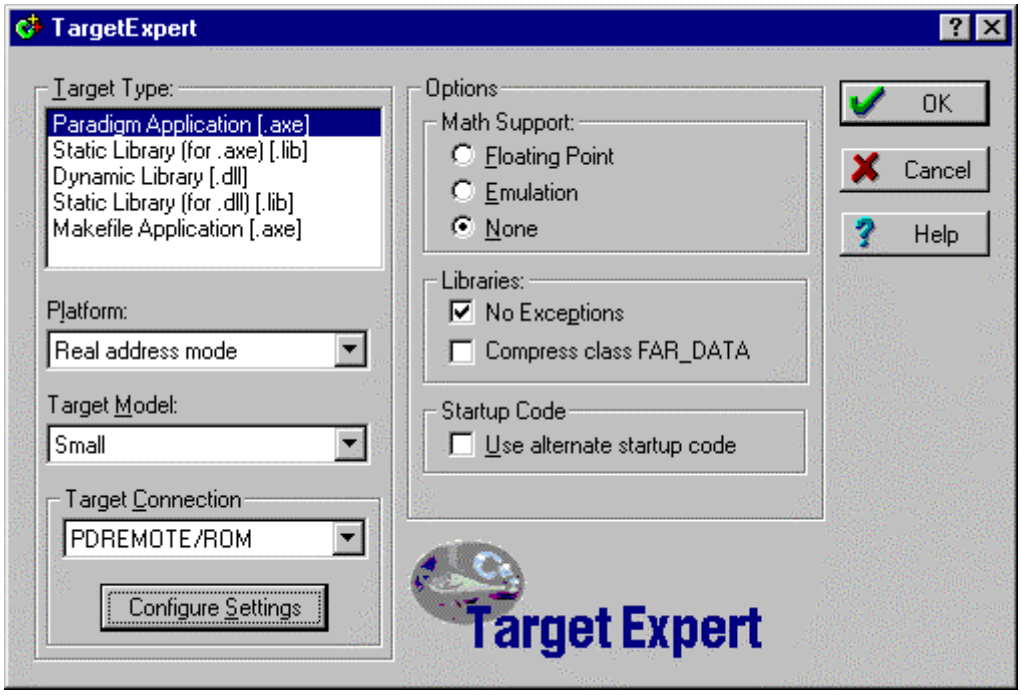


You can add more Inspect windows to see more than on variables, ledd and k.

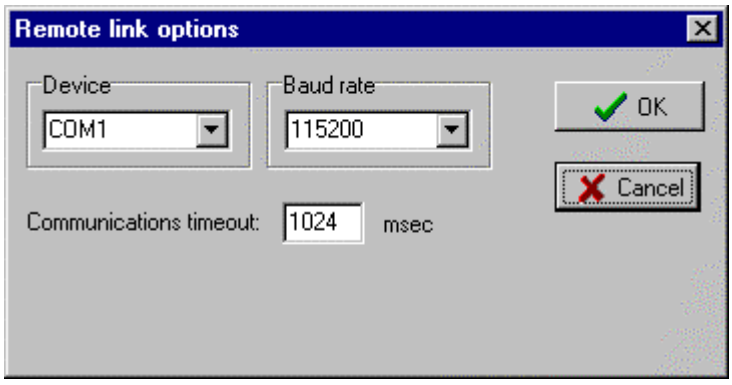


How do I change the debug baud rate ?

In side the “project window”, right click on the .axe node, select the TargetExpert.

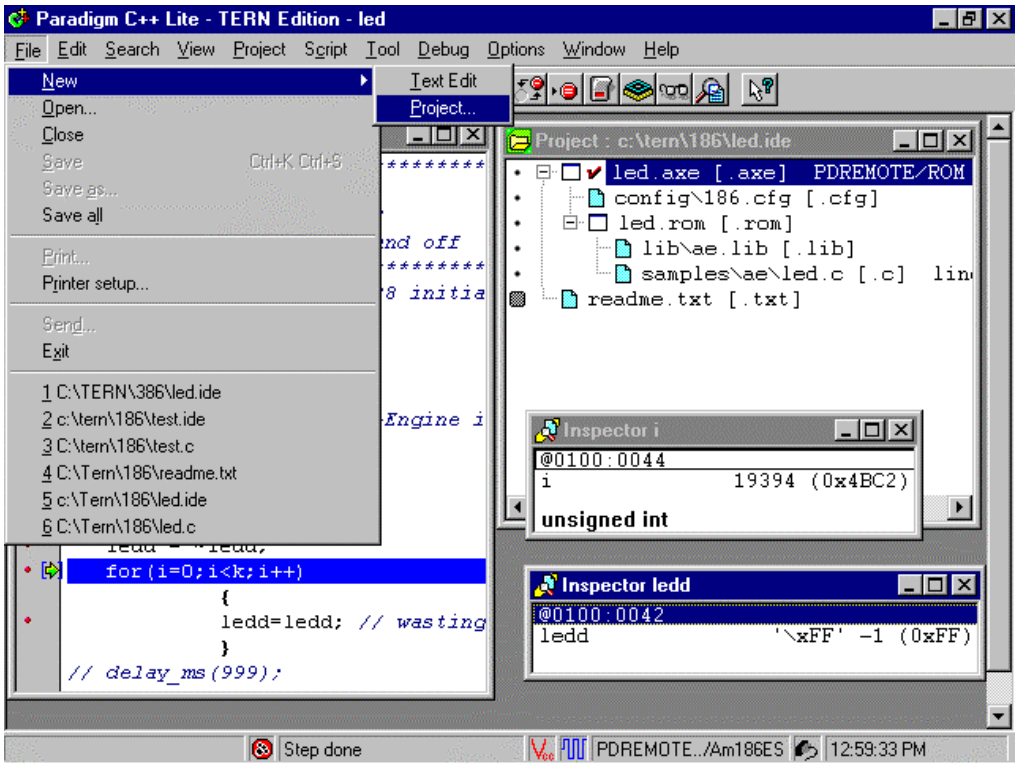


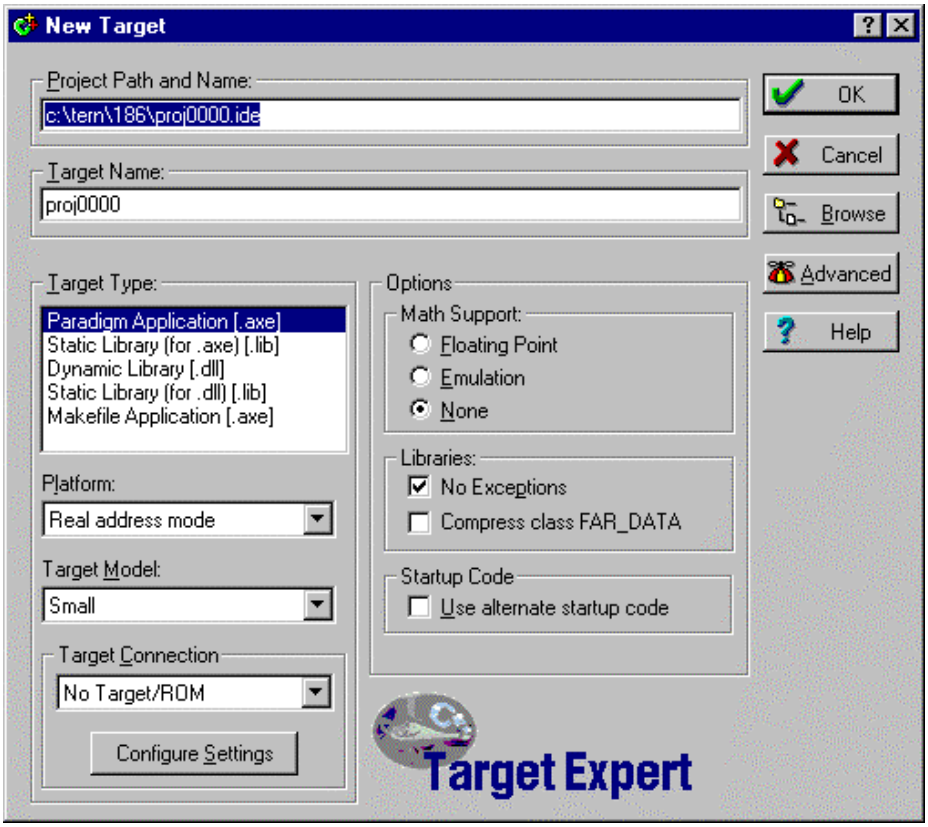
Then select the “Configure Settings”. You can select the Remote link options.

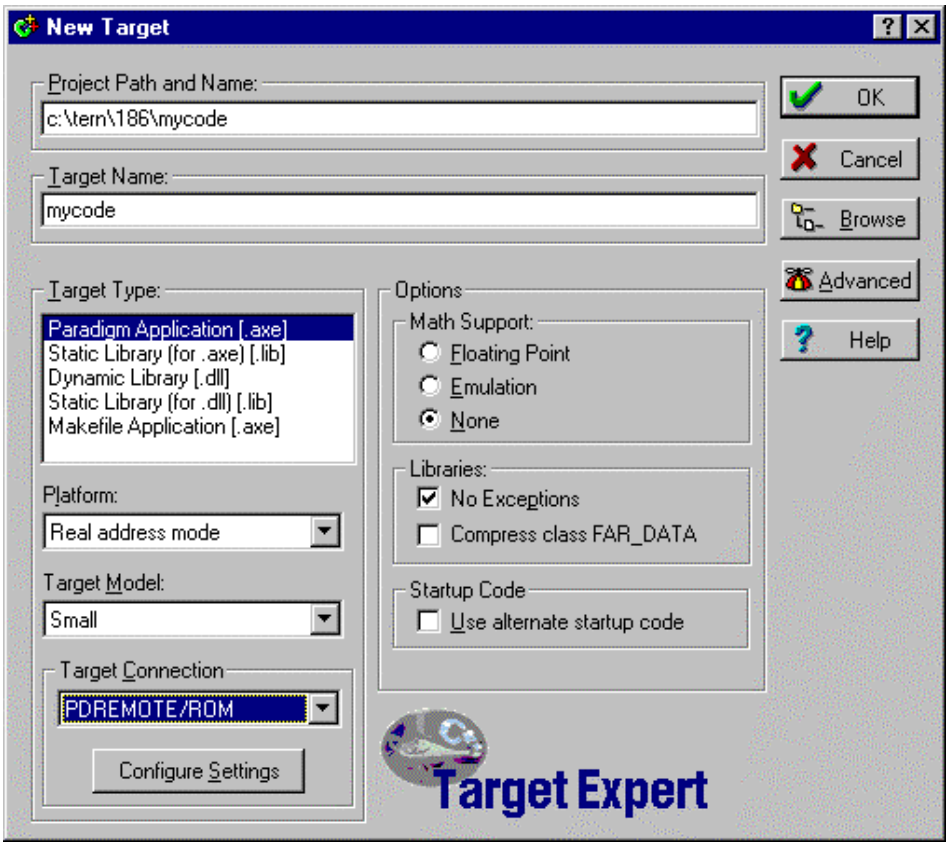


How do I make a new project ?

Select the “File”, then “new”, then “Project”, a “New Target” window pops up.

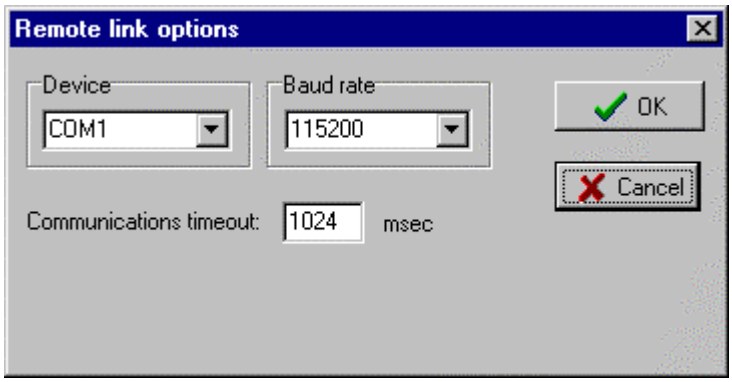






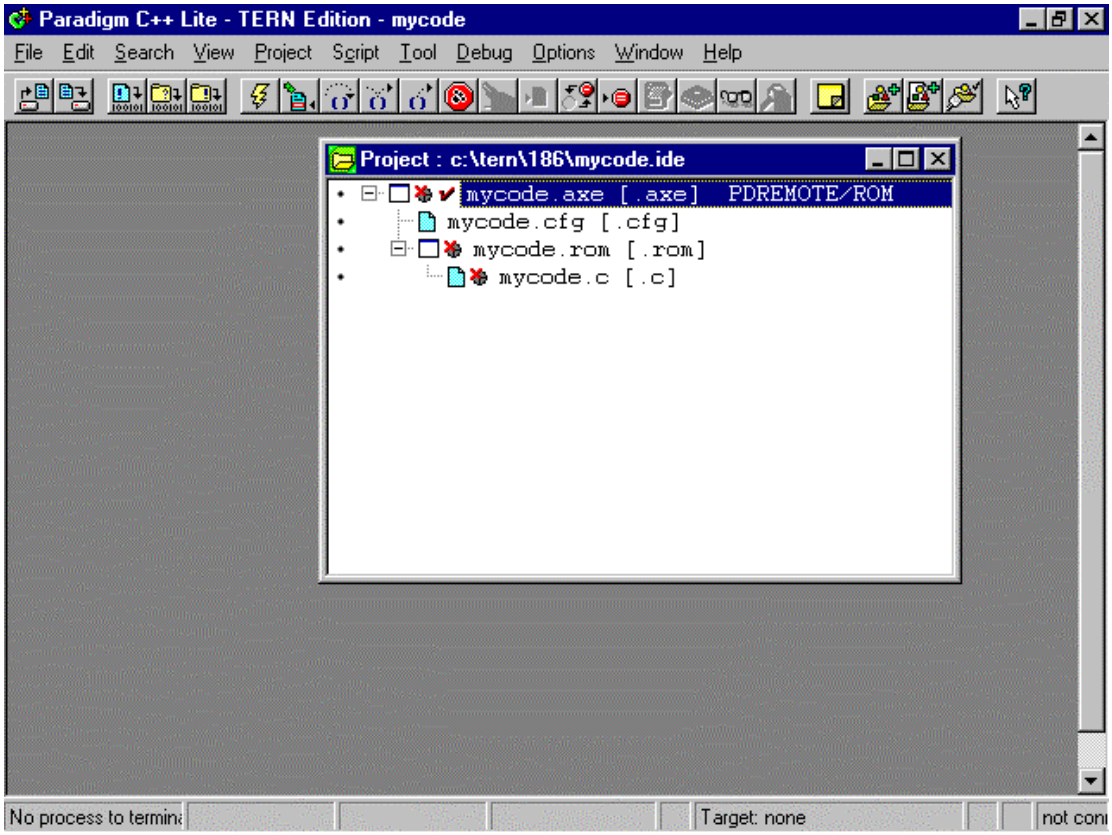
You need to setup the “Project Path and Name” to c:\tern\186\mycode, as an example for a TERN Am188 based controller and your application “mycode”.

Always make sure the Libraries: No Exceptions is **CHECKED**. You also need to define the Target Connection as PDREMOTE/ROM for debugging. If you select No Target/ROM, in order to generate a ROM file, then you have to have the full version Paradigm C++-TERN Edition, not the Lite evaluation version.



Select the Configure Settings to setup correct Remote link options.

A new project named c:\tern\186\mycode.ide is created.



What is a node ?

In the Project window, each line can be called a “node”. The root node is always the [.axe] node, which representing the final application code ready for downloading to your controller.

The [.axe] node has two children nodes: [.cfg] and [.rom].

The generation of the [.axe] nodes depends on the [.cfg] node and the [.rom] node.

The [.cfg] node is the configuration file for mapping your code and data. The [.rom] node is the product of the application source compiled and linked by C compiler and linker.

A default configuration file is under c:\tern\186\config\186.cfg. You may use it by change the name of the “mycode.cfg”.

The mycode.rom [.rom] node has its own children. The numbers of children nodes depends on your application including all source files and necessary libraries.

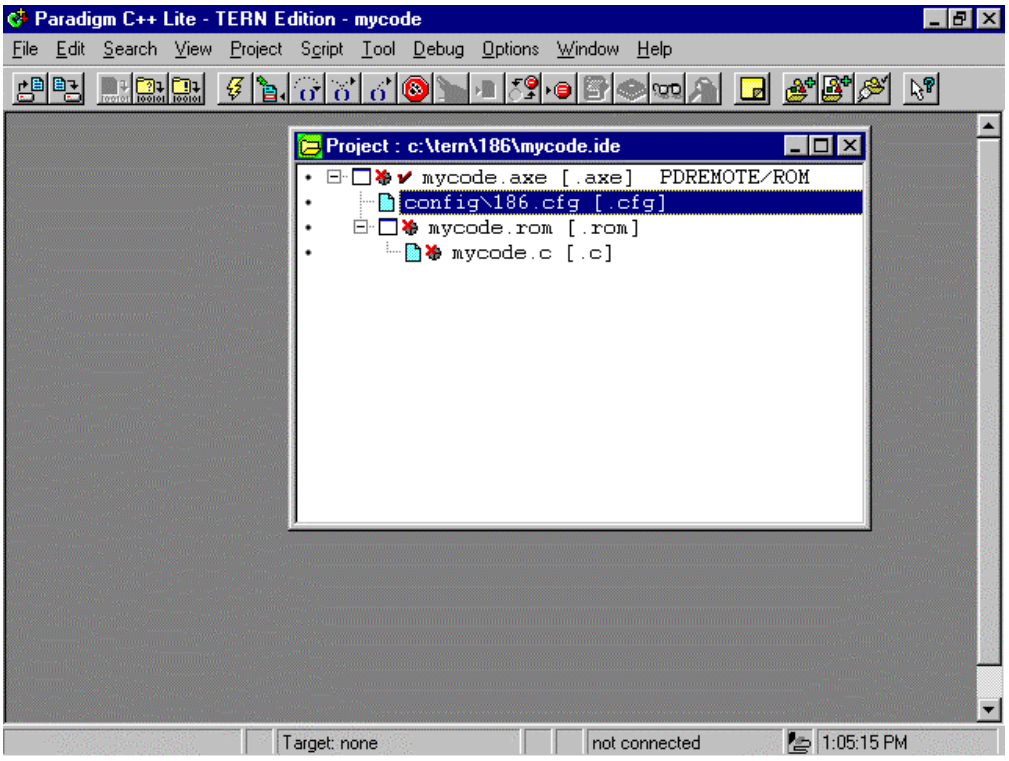
Only the source code “mycode.c” may be not enough. You may need to add a children node of “c:\tern\186\lib\ae.lib” for example.

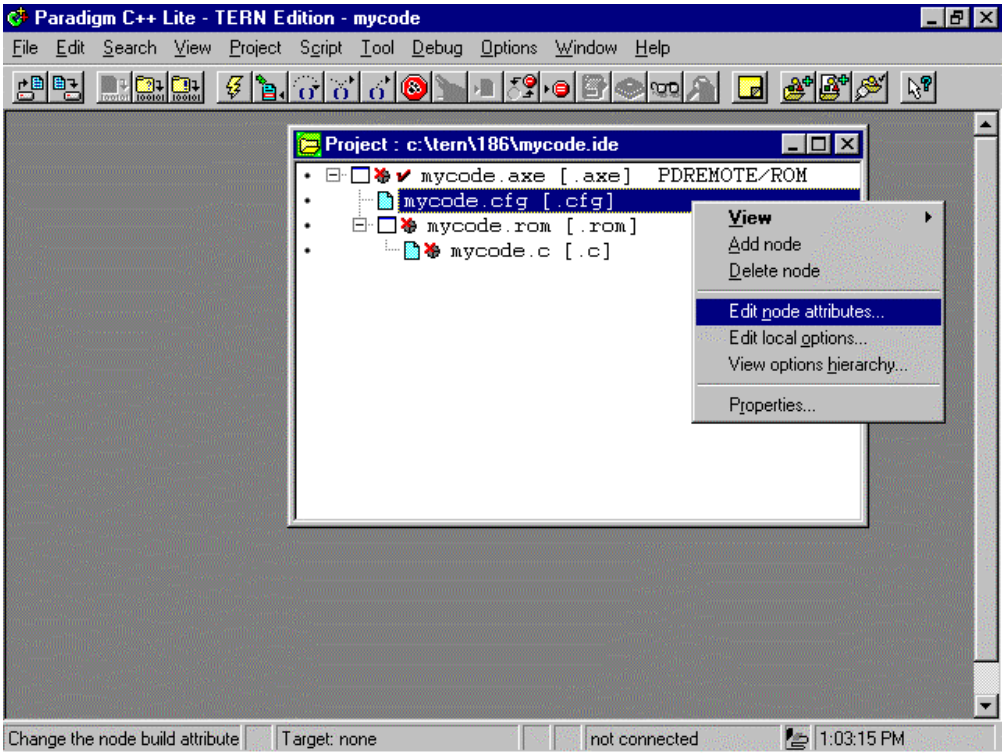
How to move a node in the Project window ?

Highlight the node you want to move, Hold down the “Alt” key, then use up or down arrow.

How to change or rename, or delete a node ?

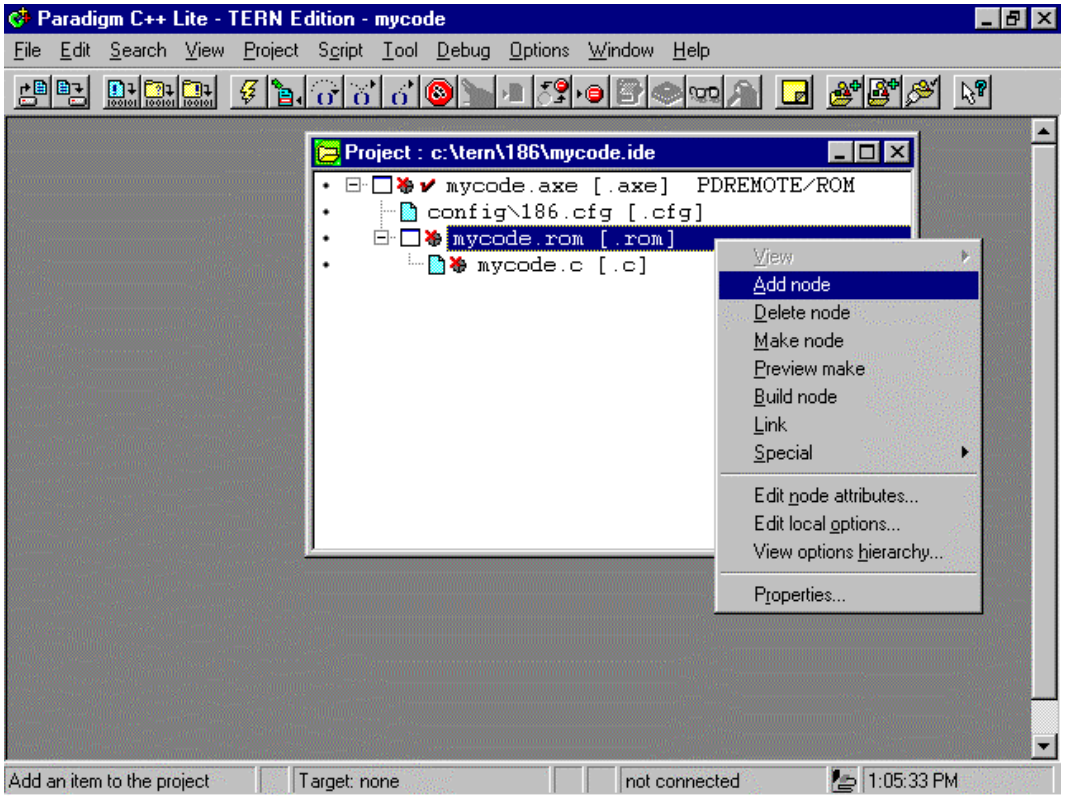
Lets try to rename the mycode.cfg node to config\186.cfg. highlight the node, right click.



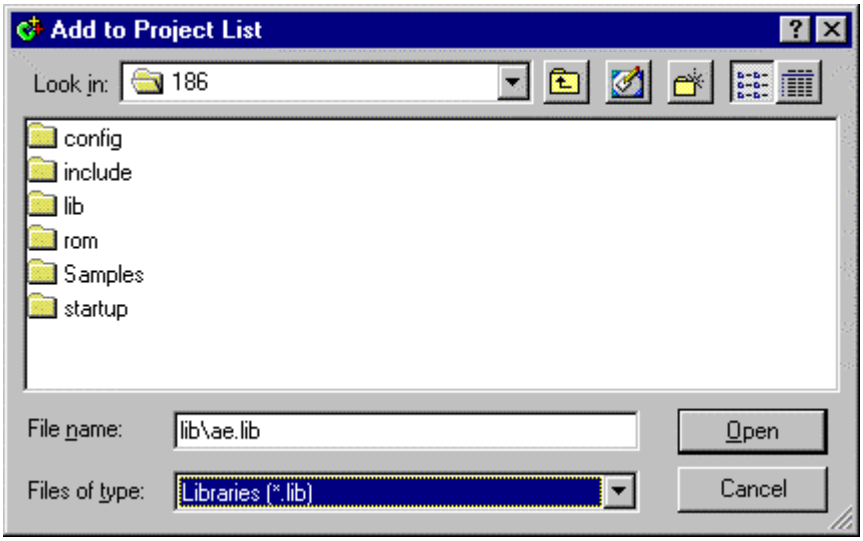


How do I add a node, ae.lib for example.

Select the father node of the new children node you are going to add.



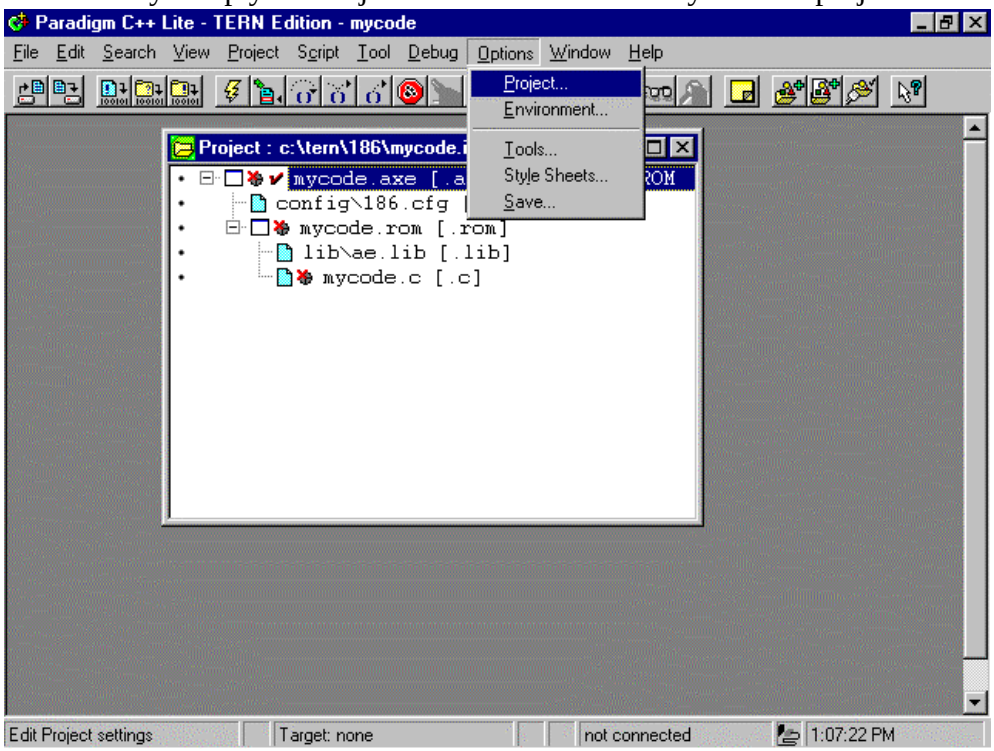
For example, if you want to add a ae.lib node for the mycode.rom [.rom] node, you highlight the mycode.rom [.rom] node, then right click, then select “Add node”.
A window of “Add Project List” pops up.



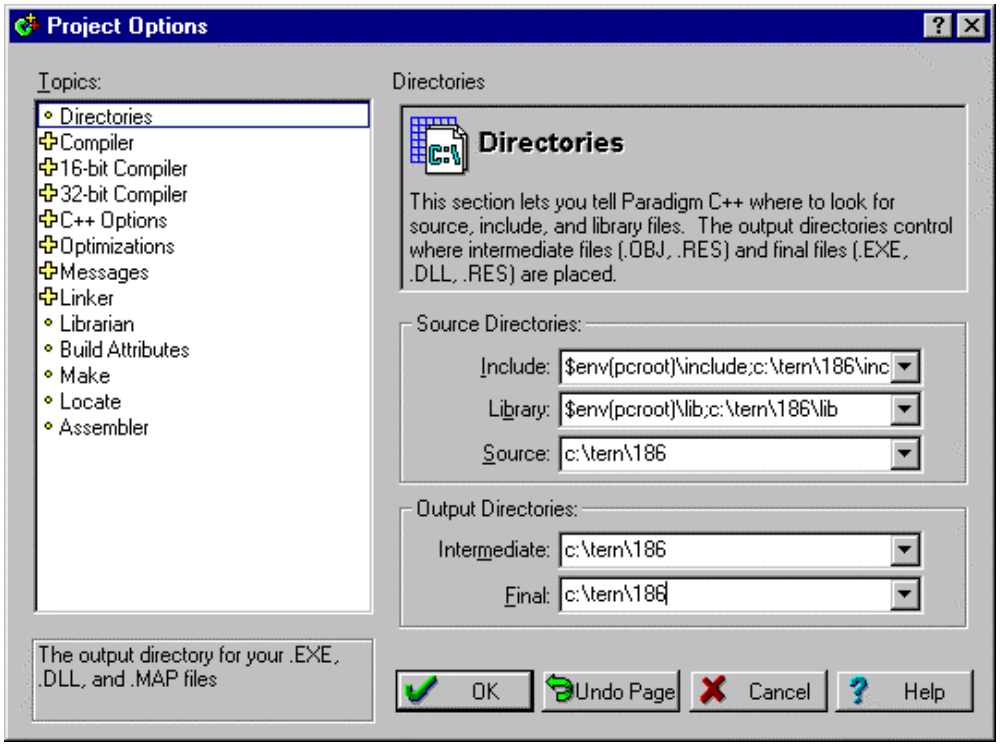
You can type in the lib\ae.lib, or use mouse to select.

How do I setup the Project and Environment ?

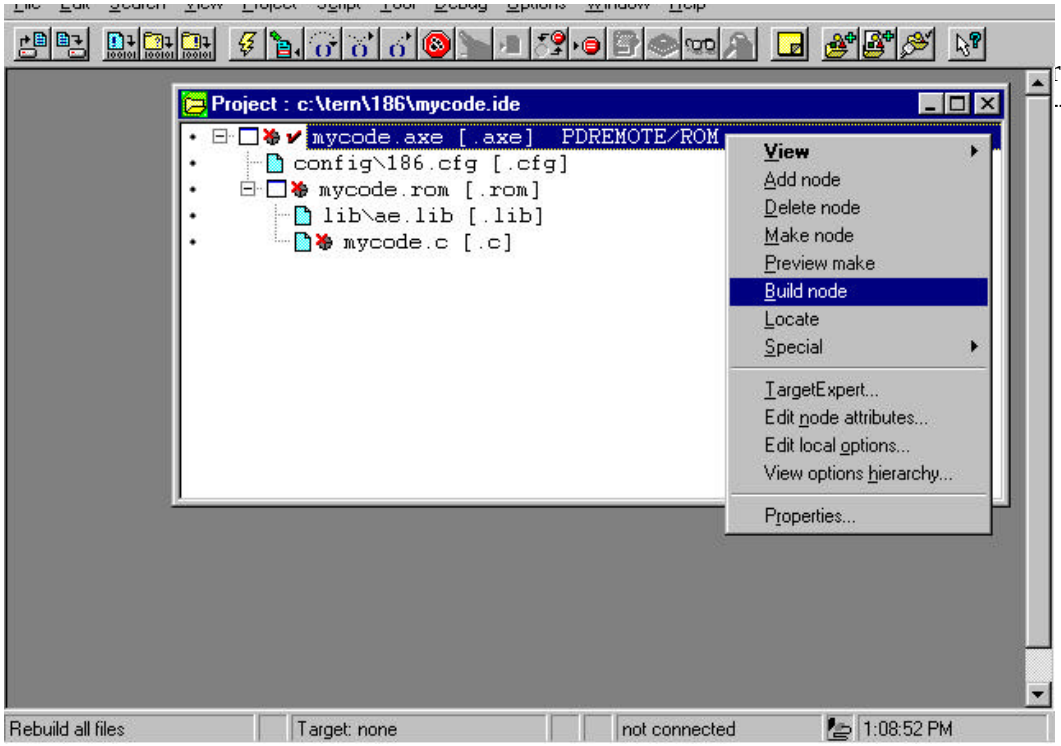
You must correctly setup your Project and Environment for your new project.



At the top line menu, select Options, then Project. A Project Options window pops up.

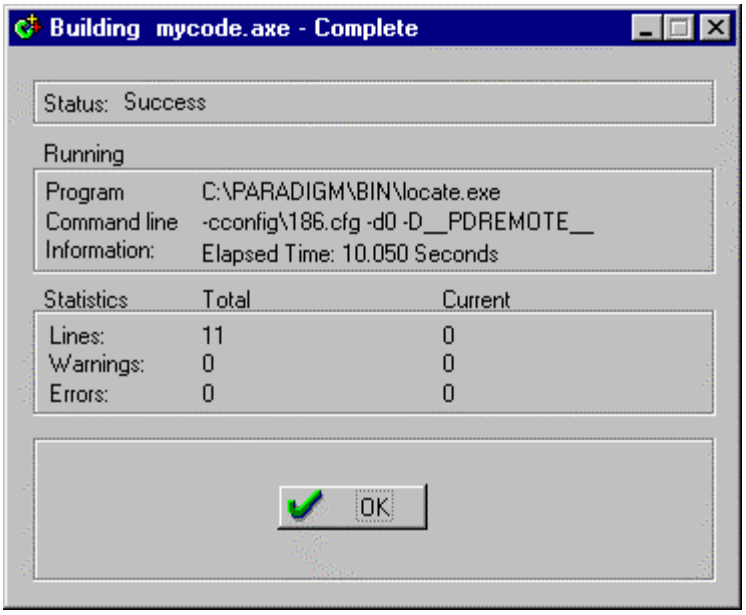


You must add the working directory to Include, Library and source, as shown above.



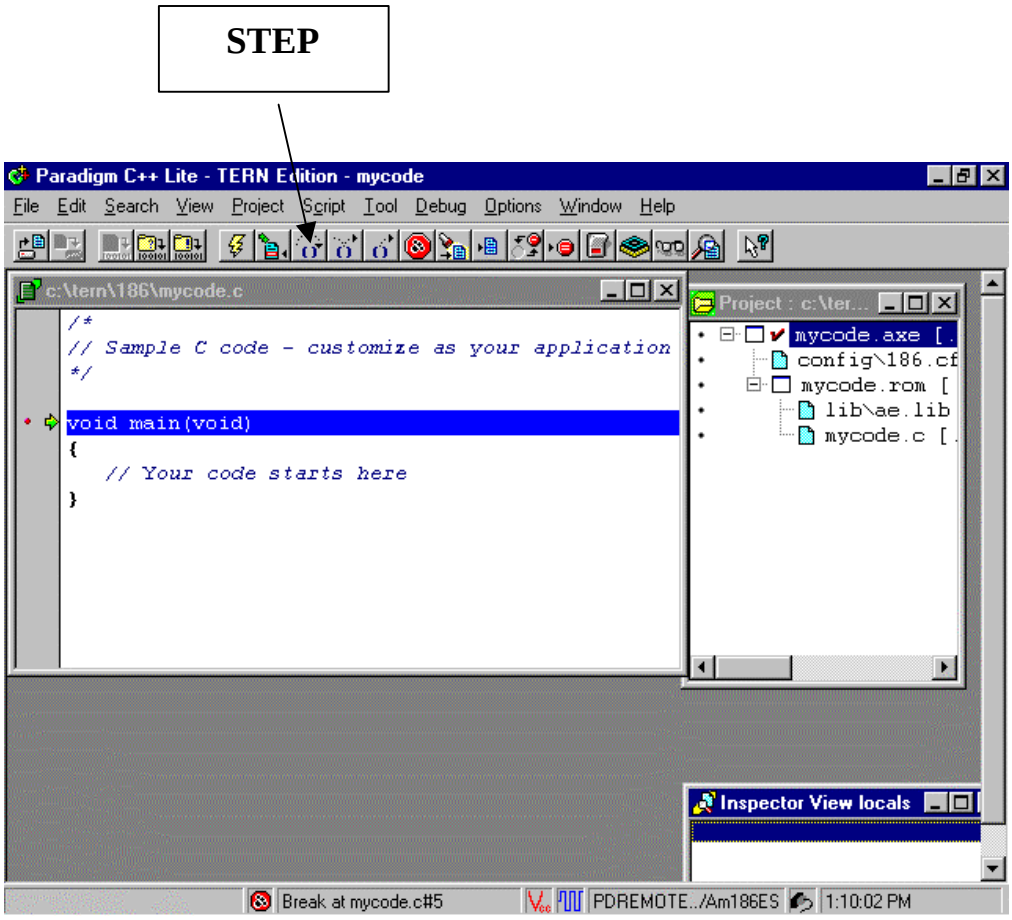
How do I test my project and environment setting ? Build node.

Any time, you can select the [.axe] node, then right click, select “Build node”. It will compile and link all your children nodes under the [.rom] nodes. Produce .[rom] file, then locate to correct memory mappin based on the configure file in the [.cfg] node, finally generate the [.axe] file.



How do I download the new project code ?

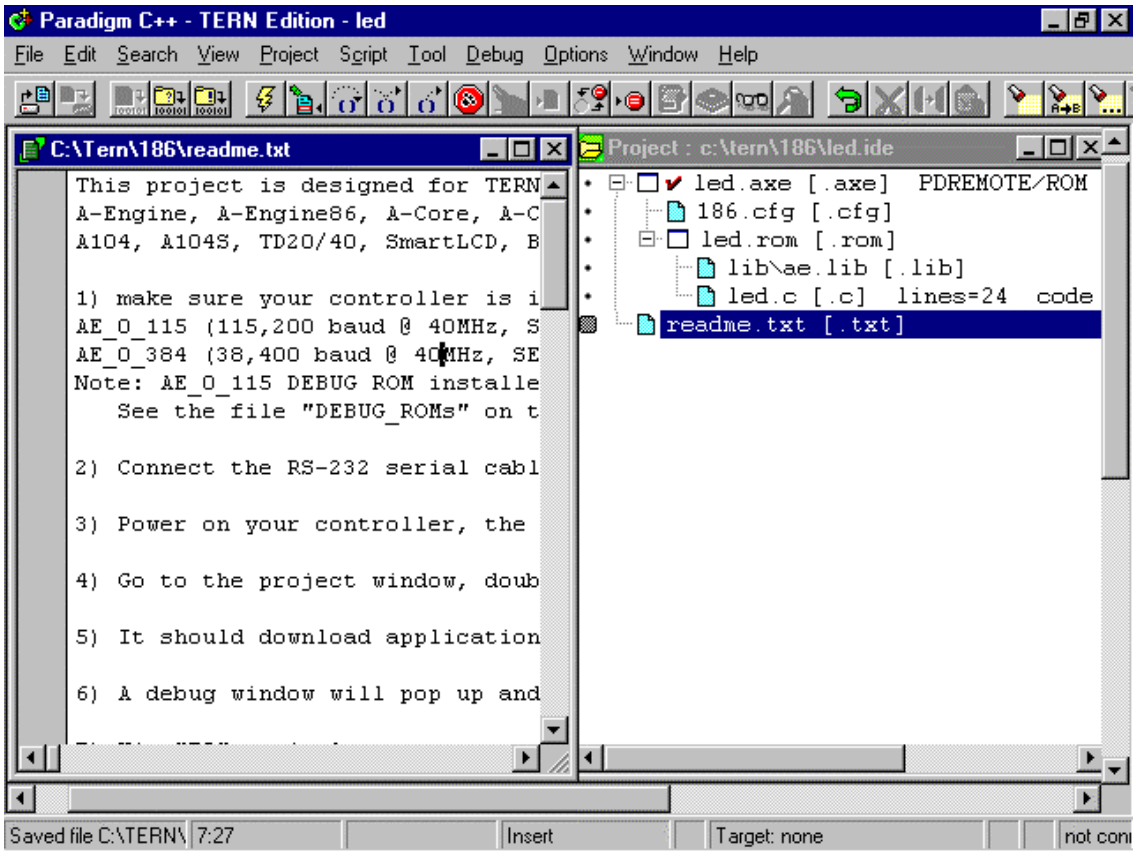
Right click on the STEP button, it will build, and download your code, ready for you to debug, and run



How to setup and monitor a variable or variables ?
How to stop a program ?
How to terminate a debug session ?

How to open a file ?

Right double click on the readme.txt [.txt], it will open it for you.



Chapter 4: Questions & Answers

4.1 How does the software organize on my PC?

After install the Paradigm C++-TERN Edition CD, directories are created on your PC drive C: for programming TERN embedded controllers.

TERN controllers are based on 80x86 PC compatible microprocessors, such as V25 from NEC, Am188/186 from AMD, and i386EX from Intel. Embedded hardware drivers are provided in libraries. V25 based files are under c:\tern\v25. Am186 based files are under c:\tern\186, i386EX based files are under c:\tern\386.

Core sample programs for controllers are in sub-directories:
c:\tern\186\samples\ae for Am188/186, c:\tern\386\samples\ie, and c:\tern\v25\samples\ve.

```
After the installation of the CD, the directory C:\TERN and sub-
directories are created.

C:\TERN\V25  ------- working directory for V25 based controllers
C:\TERN\V25\SAMPLES\ ---- sample programs
C:\TERN\V25\CONFIG\ ----- configuration files
C:\TERN\V25\INCLUDE\ ---- Header files
C:\TERN\V25\LIB\ -------- Libraries for V25 boards

C:\TERN\186  ------- working directory for Am188/6 based controllers
C:\TERN\186\SAMPLES\ ---- sample programs
C:\TERN\186\CONFIG\ ----- configuration files
C:\TERN\186\INCLUDE\ ---- Header files
C:\TERN\186\LIB\ -------- Libraries for 186 boards

C:\TERN\386  ------- working directory for i386EX based controllers
C:\TERN\386\SAMPLES\ ---- sample programs
C:\TERN\386\CONFIG\ ----- configuration files
C:\TERN\386\INCLUDE\ ---- Header files
```
C:\TERN\386\LIB\ ---------- Libraries for 386 boards

C:\Paradigm\ ---------- Paradigm C++-TERN Edition files.

4.2 Where is the initial working directory?

If you are using V25 based controllers, your working directory is c:\tern\v25.
If you are using 186 based controllers, your working directory is c:\tern\186.
If you are using 386 based controllers, your working directory is c:\tern\386.

4.3 What is the Project file?

A project file is an intelligent program manager that does all the work necessary to keep your program up-to-date. It reads a special “makefile” designed for managing whole compile, link, debug and locate operation.
The project file uses .ide.

4.4 Controller Memory mapping

There is 1 MB memory space for 16-bit controllers. TERN controllers map lower 512KB (0x00000-0x7ffff) to the RAM socket, and map upper 512KB into ROM socket.
In order to correctly download a program in STEP1 via PC RS232 serial link, the controller must meet these requirement:

1). DEBUG EPROM "AE_0_115" for A-Engine, SER0 debug at 115,200 installed in ROM socket.
See attached DEBUG ROM list

2). SRAM installed must be large enough to hold your program.
For a 32K SRAM, the physical address is 0x00000-0x07fff
For a 128K SRAM, the physical address is 0x00000-0x01ffff
For a 512K SRAM, the physical address is 0x00000-0x07ffff

3). Your program must be located at a correct starting address to match your SRAM size.
For example, if you have a 32K SRAM, which maps into 0-0x07fff. Even for a 10K program, if you locate the code into a starting address of 0x08000, it cannot be download nor run because there is no RAM.

4). Your program is located by configure file,
c:\tern\186\config\186.cfg, or
c:\tern\v25\config\v25.cfg, or
c:\tern\386\config\386.cfg.

5). In the "186.cfg" file, you will see these lines:

```
map     0x00000 to 0x00fff as reserved      // interrupt vector table
map     0x01000 to 0x07fff as rdwr          // System RAM area(32 K)
map     0x08000 to 0x1ffff as rdonly        // Simulated ROM area(96K)
map     0x20000 to 0xfffff as reserved      // No access
dup     DATA ROMDATA                        // Make a copy of initialized data
class   CODE = 0x0800           // Assume loading at address 080000H
class   DATA = 0x0100           // Data at address 01000H
```

The bolded 0x8000 indicates your code will be located at a start address of 0x0800:0000.

6). It is very important to know, what size SRAM you have, where is your effective SRAM address, and where you located your program while you are using the "186.cfg" file.
If you have a only 32K SRAM (0-0x7fff), you may modify the "186.cfg" file to select RAM=32.

```
map     0x00000 to 0x00fff as reserved      // interrupt vector table
map     0x01000 to 0x03fff as rdwr          // System RAM area(16 K)
map     0x04000 to 0x07fff as rdonly        // Simulated ROM area(16K)
map     0x08000 to 0xfffff as reserved      // No access
dup     DATA ROMDATA        // Make a copy of initialized data
class   CODE = 0x0400           // Assume loading at address 04000H
class   DATA = 0x0100           // Data at address 01000H
```

The bolded 0x4000 indicates your code will be located at a start address of 0x0400:0000.

7). In order to run STEP2 stand alone mode, you must know where your code are located during STEP1 downloading. You must correctly set the STEP2 jumping address in your serial EEPROM (EE address 0x10 to 0x13). You may run the "step2.c" program to setup your EE jumping address. You only need to run the "step2.c" once. Please see "step2.c" in c:\tern\186\samples\ae directory.

4.5 DEBUG baud rates

Paradigm C++-TERN Edition supports one of the debug Baud Rate: 19,200, or 38,400, or 57,600, or 115,200.
The DEBUG ROM installed determines the DEBUG Baud Rate.
For example: DEBUG ROM labeled “AE_0_115” installed on a 40 MHz A-Engine, the debug baud rate is 115,200. The same debug rom installed on a 20 MHz A-Engine, the baud rate is 57,600.

4.6 What are differences between the Lite version and full version?

The Lite version supports STEP1 and STEP2 only, not STEP3.
In addition to all the Lite version features, the full version provides support that you will need to generate your own application ROM, STEP3.

4.7 What is ACTR/ACTF™?

Operating TERN's 16-bit controllers with ACTR/ACTF™

With ACTR™/ ACTF™ firmware support, you only need a terminal, such as PC windows terminal, set up to 19200, 8, N, 1, via serial link. By typing text commands, you can operate the controller and exercise C functions immediately. The ACTR/ACTF™ provides an interactive menu and on-line help for you, so you do not have to dig into manuals. ACTR/ACTF™ not only provides you easy access to C functions, but also allows you to download a remote debugger kernel for your C/C++ program development.
ACTR™ is in a 64K EPROM. ACTF™ is a FLASH version of ACTR™. You may use DV kit to generate your application HEX file and download the file into the on-board Flash with ACTF™.
ACTF™ can be used in your final OEM product. As long as the jumper is on every time the power is on or reset, the controller always runs your program. You may download several application programs in different memory locations, remove the jumper, use ACTF™ “***G***xxxxx” commands to set up a new CS:IP, and when the jumper is restored, it will run that program at power on or reset

4.8 Where is the technical manual?

Complete sets of Technical Manuals, photos, schematics, and help manuals are in the Paradigm C++-TERN Edition CD. You may use Windows Explore to copy, use Adobe Reader to read and print.

4.9 What is the .MAP file and the .LOC file?

After you “Build node” on the .axe [.axe] node, or run the code on the controller it will generate a .map file and a .loc file, in your working directory, c:\tern\186.
The led.map file, as below, will list generated code and data locations.

```
Start   Stop   Length Name               Class

 00000H 0187DH 0187EH _TEXT              CODE
 01880H 01897H 00018H _INIT_             INITDATA
 01898H 01898H 00000H _INITEND_          INITDATA
 01898H 01898H 00000H _EXIT_             EXITDATA
 01898H 018A7H 00010H _EXITEND_          EXITDATA
 018B0H 018B0H 00000H _RD                ROMDATA
 018B0H 018BFH 00010H _ERD               ENDROMDATA
 018C0H 01A43H 00184H _DATA              DATA
 01A44H 01A44H 00000H _CVTSEG            DATA
 01A44H 01A44H 00000H _SCNSEG            DATA
```

```
 01A44H 01CA1H 0025EH _BSS                BSS
 01CA2H 01CA3H 00002H _BSSEND             BSSEND
 01CB0H 02CAFH 01000H _STACK              STACK
 02CB0H 02CB0H 00000H _BFD                FAR_DATA
 02CB0H 02CBFH 00010H _EFD                ENDFAR_DATA
 02CC0H 02CC0H 00000H _BRFD               ROMFARDATA
 02CC0H 02CCFH 00010H _ERFD               ENDROMFARDATA

Program entry point at 0000:0000
```

The led.loc file, as below, will list generated code and data locations, which is the memory location on the target controller.

```
Memory Address Map for Program LED

 Start     Stop      Length   Segment              Class
 001000H   001183H   00184H   _DATA                DATA
 001184H   001184H   00000H   _CVTSEG              DATA
 001184H   001184H   00000H   _SCNSEG              DATA
 001184H   0013E1H   0025EH   _BSS                 BSS
 0013E2H   0013E3H   00002H   _BSSEND              BSSEND
 0013F0H   0023EFH   01000H   _STACK               STACK
 0023F0H   0023F0H   00000H   _BFD                 FAR_DATA
 0023F0H   0023FFH   00010H   _EFD                 ENDFAR_DATA
 008000H   00987DH   0187EH   _TEXT                CODE
 009880H   009897H   00018H   _INIT_               INITDATA
 009898H   009898H   00000H   _INITEND_            INITDATA
 009898H   009898H   00000H   _EXIT_               EXITDATA
 009898H   0098A7H   00010H   _EXITEND_            EXITDATA
 0098B0H   0098B0H   00000H   _RD                  ROMDATA
 0098B0H   009A33H   00184H   _DATA                ROMDATA
 009A34H   009A34H   00000H   _CVTSEG              ROMDATA
 009A34H   009A34H   00000H   _SCNSEG              ROMDATA
 009A40H   009A4FH   00010H   _ERD                 ENDROMDATA
 009A50H   009A50H   00000H   _BFD                 ROMFARDATA
 009A50H   009A50H   00000H   _BRFD                ROMFARDATA
 009A50H   009A5FH   00010H   _ERFD                ENDROMFARDATA

Entry point:   0800:0000
Initial stack: 013F:1000
```

4.10 EPROM and FLASH support.

Your final firmware can be program into a EPROM, such 27C256-70 (32K) or 27C512-70 (64K), with your EPROM programmer.
You can also use ACTF kit to generate a downloadable .HEX file, and install an FLASH ACTF chip in the ROM socket. With a terminal serial link at 19,200, you can download your code and program the Flash on-board.

4.11 DEBUG ROM List is on your CD, under tern_docs.

4.12 How to access 1 MB memory

Most application has been using Small Model to program TERN controllers. The Small Model code is smaller, more efficient, more reliable, and faster, but it is limited in 64K code and data. Many users developed successfully data acquisition type product with Small Model and collecting more than 64K data. The V25 or 80x86 processor supports memory access by simply using
pokeb (segment, offset, dat); and peekb (segment, offset).
Please refer to the .LOC file listed in Chapter 4, After you use “m.bat” batch file, a .LOC file will generated. It will show where and how many bytes that your C CODE and DATA will use. Suppose, your complete C stuff only use upto 0xffff, but your SRAM is 128K, mapping in 0x0000 to 0x1ffff. So you have memory space from 0x10000 to 0x1ffff of a total 64K space, which are free, battery backed, very save for you to store your data.
You may use pokeb(0x1000, unsigned int offset, unsigned char data); to store your data and increment offset, and use peekb(0x1000, unsigned int offset); to read data back in the future.

Appendix A: Debug ROM list for TERN controllers

File Name	*Size*	*Lable Name*	*Debug*	*Controllers*	*Other*
TDREM_AE.BIN	32K	TDREM_AE 115,200@40M	TD31	AE/A104/TD40/BBA/SLMD88	188 SER0
TDREM_AC.BIN	32K	TDREM_AC 115,200@40M	TD31	AC	188 SER0
TD384_AE.BIN	32K	TDREM_AE 38,400@40M	TD31	AE/A104/TD40/BBA/SLMD88	188 SER0
TDREM_VE.BIN	32K	TDREM_V25 115,200@16M	TD31	VE/CE/TD/V104	V25 SER0
TDREM_IE.BIN	32K	TDREM_i386EX 115,200@66M	TD31	IE/ID/IEP 8-bit SRAM	386 SER0
TD_ID_16.BIN	32K	TD_ID_16 115,200@66M	TD31	ID 16-bit SRAM	386 SER0
TD_IE_16.BIN	32K	TD_IE_16 115,200@66M	TD31	IEP/IEM 16-bit SRAM	386 SER0
AE_0_115.BIN	32K	AE_0_115 115,200@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
AE_1_115.BIN	32K	AE_1_115 115,200@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER1
AE_0_384.BIN	32K	AE_0_384 38,400@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
AE_1_384.BIN	32K	AE_1_384 38,400@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
AC_0_115.BIN	32K	AC_0_115 115,200@40M	CPP	AC	188 SER0
AC_1_115.BIN	32K	AC_1_115 115,200@40M	CPP	AC	188 SER1
AC_0_384.BIN	32K	AC_0_384 38,400@40M	CPP	AC	188 SER0
AC_1_384.BIN	32K	AC_1_384 38,400@40M	CPP	AC	188 SER1
IE8_115.BIN	32K	IE8_0_115 115,200@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0
IE8_384.BIN	32K	IE8_0_384 38,400@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0
IE8_115D.BIN	32K	IE8_0_115 DCD1 115,200@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0 DCD1
IE8_384D.BIN	32K	IE8_0_384 DCD1 38,400@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0 DCD1
ID16_115.BIN	32K	ID16_0_115 115,200@66M	CPP	ID 16-bit SRAM	386 SER0
IE16_115.BIN	32K	IE16_0_115 115,200@66M	CPP	IEP/IEM 16-bit SRAM	386 SER0
ID16_384.BIN	32K	ID16_0_384 38,400@66M	CPP	ID 16-bit SRAM	386 SER0
IE16_384.BIN	32K	IE16_0_384 38,400@66M	CPP	IEP/IEM 16-bit SRAM	386 SER0

VE_0_115.BIN	32K	VE_0_115 115,200@16M	CPP	VE/CE/TD/V104	V25 SER0
VE_1_115.BIN	32K	VE_1_115 115,200@16M	CPP	VE/CE/TD/V104	V25 SER1
VE_0_384.BIN	32K	VE_0_384 38,400@16M	CPP	VE/CE/TD/V104	V25 SER0
VE_1_384.BIN	32K	VE_1_384 38,400@16M	CPP	VE/CE/TD/V104	V25 SER1